Universidade do Minho
Escola de Engenharia

Filipa Alexandra Esteves de Araújo

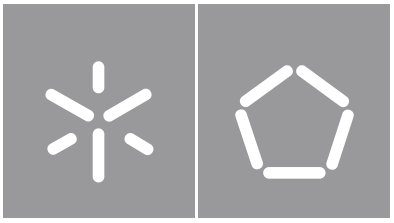
Edge of the Network Device for a Low Power Wide Area Network

January 2019

Edge of the Network Device for a Low Power Wide Area Network

Filipa Araújo

UMinho | 2019



Universidade do Minho
Escola de Engenharia

Filipa Alexandra Esteves de Araújo

Edge of the Network Device for a Low Power Wide Area Network

Dissertação de Mestrado
Engenharia Eletrónica Industrial e Computadores

Trabalho efetuado sob a orientação do
Professor Doutor Jorge Cabral

January 2019

Acknowledgements

I would like to thank Professor Jorge Cabral for giving me the opportunity of developing this project, and for all the guidance throughout the development of this work. Thank you to everyone from the ESRG lab in IBS, for the tremendous support given.

To my family, for always being there for me every step of the way. To my mom, for all the incentive. A huge thanks to my dad, for always pushing me to pursue my dreams, and being my role model. To my stepmother, for always being contagiously cheerful. To my sisters, for showing me that the the most unexpected situations can turn out to be the best we have in life. To my nephew João, who has been such a joy in my life!

A huge thanks to my friends Filipa, Cunhal, Zé, Pedro and Neto, for all the good times, the advice they have given me, and for being awesome! To Sérgio and Miguel, for being my partners in this last year, and inviting me to face new challenges. To Álvaro, for all the friendship and showing me that small people can achieve great things. To João, for keeping me on track when I felt like quitting.

A big thank you to Joseph, for all the patience and care given through these years. You are able to get the best of me, even when I don't believe in myself. Thank you for always being there for me, and being my greatest supporter.

To Bobby, Shelah and Jordan, for being my second family and never ceasing to amaze me with all their kindness and love.

Thank you to everyone who has crossed paths with me and contributed to making me the person I am today!

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Resumo

A proliferação da conexão à *Internet*, especialmente em pequenos dispositivos (sistemas embebidos), permitiu o desenvolvimento do conceito *Internet of Things* (IoT), devido à possibilidade de ligação destes a micro serviços *web* (*Cloud*), tendo um papel crucial no desenrolar da Indústria 4.0 [1]. Tendo como principal impulsionador o avanço tecnológico das redes sem fios, foi possível ligar estes dispositivos à *Internet*, tornando-os acessíveis em qualquer lado. Assim, surgiram as *Wireless Sensor Networks* (WSNs), através da utilização de redes de dispositivos independentes (nós ou *edge devices*), equipados com sensores e atuadores, possibilitando a recolha de informação sobre o meio onde estão colocados [2].

A crescente necessidade de cobrir áreas cada vez maiores para este tipo de redes, associada a requisitos mais exigentes de consumo energético reduzido nos dispositivos, abriu caminho para o aparecimento das tecnologias *Low Power Wide Area* (LPWA). Este tipo de tecnologias consegue alcances superiores em relação às redes sem fios convencionais (*Wi-Fi*, *Bluetooth*, entre outros), permitindo maior autonomia dos nós sensores [3], tornando-se assim ideais para a sua utilização em áreas alargadas.

As recentes tragédias de incêndios que ocorreram em Portugal, em particular nos anos de 2017 e 2018, tiveram grande impacto tanto a nível económico como social. A deteção e alerta precoce de incêndios são fatores cruciais para evitar a sua propagação [4]. Utilizando as tecnologias LPWA em contexto florestal poderá criar-se um caso de estudo para a ocorrência de incêndios em florestas. Através da utilização de *edge devices*, poderá ser possível recolher dados provenientes deste meio que indiquem a existência de um incêndio a deflagrar, e enviar alertas para as unidades de combate a incêndios.

Nesta dissertação foi desenvolvida a arquitetura dos nós sensores, a serem integrados numa *Low Power Wide Area Network* (LPWAN). Utilizando tecnologia LoRa para obter um longo alcance entre os nós e o coordenador da rede, poderá desta forma ser possível os nós sensores recolherem e enviarem dados para as camadas superiores.

Foi possível, com a utilização de sensores nos nós, recolher informações sobre o ambiente e perceber o potencial da tecnologia LoRa para o envio destes dados para as camadas superiores.

Palavras-chave: *Low Power Wide Area*, Redes de sensores, Nó sensor, LoRa.

Abstract

The widespread of Internet connection, particularly on small devices (embedded systems), has allowed the development of the Internet of Things (IoT) concept, due to the connection of these devices to web micro services (Cloud), and has had a major role in Industry 4.0 [1]. Through the advances of wireless technologies, these devices were able to have an Internet connection, becoming available everywhere. The creation of Wireless Sensor Networks (WSNs) has enabled the use of networks composed of independent devices (nodes or edge devices), equipped with sensors and actuators, and made it possible to collect information about the environment where they are deployed [2].

The growing necessity of having a wider coverage area for Wireless Sensor Networks, along with the demanding low power requirements on devices has enabled Low Power Wide Area (LPWA) technologies to arise. These technologies are able to reach further coverage than conventional wireless technologies (such as Bluetooth, Wi-Fi, ZigBee etc), as well as raising the energy autonomy of the devices [3], which makes LPWA technologies ideal for wider areas.

The recent tragedies of wildfires in Portugal, in both 2017 and 2018, had great impact on economic and social levels. Early detection and alerts about wildfires are crucial to prevent them from spreading [4]. Therefore, by using LPWA technologies in forests, a case study can be made for the wildfire occurrences in forests. Through the use of independent devices equipped with sensors, data can be collected from the environment that might detect that a fire is starting, and then send alerts to fire fighting units.

In this *Master's thesis* it was developed the architecture of sensor nodes, to be integrated in a Low Power Wide Area Network (LPWAN). By using the LoRa technology to achieve a long range between the sensor nodes and the network coordinator, it is possible for edge devices to collect and send data to upper levels of the network.

It was possible to gather information about the environment and further understand LoRa's potential for sending all the data to the upper levels of the network.

Keywords: Low Power Wide Area, Wireless Sensor Networks, Edge Device, LoRa.

Table of Contents

List of Figures

List of Tables

List of Listings

List of equations

Acronyms List

ABP Activation By Personalization.

ADC Analog to Digital Converter.

ADR Adaptive Data Rate.

AES Advanced Encryption Standard.

API Application Programming Interface.

BPSK Binary Phase-Shifting Key.

CPS Cyber Physical System.

CRC Cyclic Redundancy Check.

EUI Entexded Unique Identifier.

HAL Hardware Abstraction Layer.

IDE Integrated Development Environment.

IoT Internet Of Things.

IP Ingress Protection.

LoRa Long Range.

LPWA Low Power Wide Area.

LPWAN Low Power Wide Area Network.

LTE Long Term Evolution.

MCU Microcontroller Unit.

MIC Message Integrity Check.

NDIR Non-Dispersive Infrared.

NTC Negative Temperature Coefficient.

OTAA Over-The-Air Activation.

PTC Positive Temperature Coefficient.

RPMA Random Phase Multiple Access.

RTD Resistance Temperature Detector.

RTOS Real Time Operating System.

SF Spreading Factor.

SWD Serial Wire Debug.

TTN The Things Network.

TxPower Transmission Power.

WSN Wireless Sensor Network.

Chapter 1

Introduction

The wide spread of Internet connection occurred not only on conventional computers, but also on small devices. In this context, the Internet Of Things (IoT) came to life [5], which is based on small devices being connected between themselves and the Internet. Therefore, any small device equipped with sensors can collect and share information, as well as interact with the environment. When combined, they are part of smart networks that allow innovative and flexible systems, whose purpose is to monitor, and manage their own devices automatically. These networks allow better human-machine and machine-to-machine interactions, as more devices are connected.

The IoT domain has appeared in a technological revolution that shapes the future of computation and communication, and its development is fully dependent on other innovations, such as sensors and wireless technologies. The IoT system architecture defines how the edge devices interact with each other and the upper levels. As it is not a specific concept, it is hard to find a convention for the network architecture [6]. However, in order to be functional, an IoT system must have sensors, a network, communication, and computational technologies. Along with these, there may also be a number of services and a web application, which can serve as a user interface.

The groups of devices are heterogeneous when it comes to their own resources, lifetime, and communication technologies. This allows IoT systems to be based on a Service-Oriented Architecture (SOA), which is a collection of services that communicate with each other. This type of model makes it possible to insert and interact with the systems that contain heterogeneous devices, making the integration with new technologies easier. As a result, it is possible to rely on Cloud services to give the abstraction level that would otherwise require a large set of computational power. These Cloud services provide processing functionalities, storage capacity, and the integration of services, thus allowing the devices that collect data to have lower processing power

requirements and be more low power oriented. This makes their deployment easier, while also reducing maintenance needs due to extended battery life.

It is due to these integrations of technologies, Cloud, and services, that the IoT has grown as one of the major fields of research and development, with a vast amount of devices being deployed.

1.1 Contextualization

All of these new concepts have brought many developments related to connections between devices and have introduced Industry 4.0, which is heavily based on the IoT concept. Wireless technologies have allowed such connections to evolve, hence being the major contributor for such evolution. This allows small devices to be connected to the Internet, which in turn opens a range of new possibilities for applications.

Nowadays, these small devices are becoming more common, allowing data to be collected from wherever they are deployed. They are used for the purposes of data collection and further decision making based on the information gathered. Not only do these small systems allow an easy placement in any spot, but also they can be low cost due to the small number of components that they have. Their connectivity to upper levels, such as cloud and web services, makes the gathering of information easily accessible. These groups of devices that gather data on the environment and proceed to send it for further data collection and processing are called Wireless Sensor Networks.

The necessity of longer signal coverage to connect small devices in these systems has brought to light the Low Power Wide Area Networks (LPWANs). This has allowed the development of networks with a long range and low power consumption for devices, making it useful for monitoring systems in remote areas. Rather than only monitoring small or enclosed areas, it is now possible to take into consideration a larger set of applications, such as addressing environmental issues. With higher range for networks, some deployments can occur in places like forests, where connectivity and energy are limited or even non existent.

Forests are undeniably imperative for air quality due to their crucial role in the transformation of carbon dioxide into oxygen. Furthermore, they also harbor several species and contain a huge amount of biological systems.

Wildfires pose a serious threat not only to forests, their wildlife, and the planet's air quality [7], but also to public safety and property [8]. They release large amounts of air pollutants, such as carbon dioxide, carbon monoxide, methane, and nitrous oxides. Besides their effects to the atmosphere, these gases are also toxic, thus having an impact on human health. Therefore, it is important to monitor and contain fire outbursts and proliferation, in order to prevent catastrophic damages.

The recurring occurrence of forest fires, especially during Summer time, has shown several issues on both their early detection and provision of relevant information to emergency services about the ongoing combustion.

The high amount of wildfires that have happened in Portugal, with some that have reached catastrophic proportions in 2017, has had a huge impact on social and economical levels. Not only have thousands of hectares of forest been devastated, but also several houses were destroyed by the flames. On the 15th of October of 2017, which was considered the worst day of wildfires in the year [9], there were more than 500 active wildfires. Taking into consideration the particular case of Pedrogão Grande in June 2017, the fast spreading of the flames was due to the weather and natural elements, along with poorly made decisions by the emergency services that were caused by failures in the existing communications network [10].

The technology used in the emergency services communications network is based on 2G technology [10], which is rather obsolete when compared to more recent alternatives such as 3G and 4G. Moreover, it has proved itself inefficient in fire occurrences, as several of this network's infrastructures were destroyed by the flames. This has left the emergency operatives on the field without any access to this communication service, and consequentially without useful information about fire activity.

1.2 Motivation

The recent scenarios of wildfire occurrences have highlighted the lack of efficiency in early detection of fire outbursts, thus making it difficult to have fast interventions and to avoid fire propagation. Therefore, it is deemed necessary to find a new technological renovation for these scenarios. Early detection and warnings of forest fires can be heavily supported by exploring recent technologies.

Making use of Low Power Wide Area (LPWA) technologies can enable the possibility to create a monitoring system for remote areas such as forests and gather information on physical variables. By processing the acquired data through the use of smart algorithms, it might help detect the start of wildfires and to make predictions on which days are more prone to fire activity. This greatly improves the reliability of these systems, which can provide a better insight on the situation to emergency services.

This dissertation comes from the need to create a Low Power Wide Area Network, which is divided into three layers: sensor nodes (edge devices), network coordinator (gateway), and web services. This network is aimed for forest environments, in order to monitor and detect possible wildfire outbursts in their early stages of ignition.

The edge devices should be equipped with several sensors that monitor physical variables and send this data to the coordinator. The coordinator will manage all the information that comes from all the nodes that are part of the network. The web services will store all the network data, from the hardware specifications of the devices themselves to the environment data collected by them. They will then analyse all of the variables, which will then provide reliable knowledge of the area and will allow a better intervention in the occurrence of a fire.

This three layer division, which correspond to three different dissertations, is properly represented in Figure 1.1.

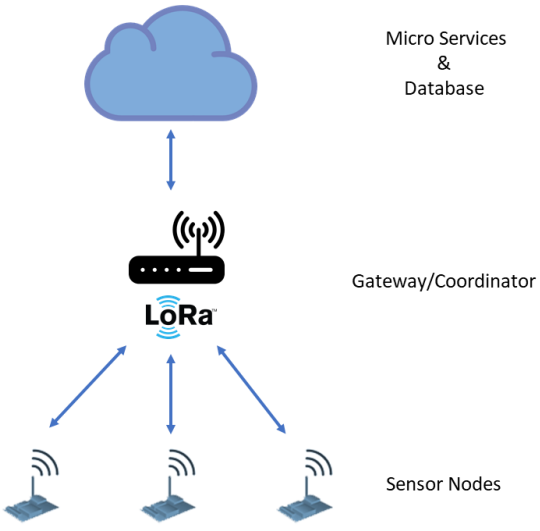


Figure 1.1: Division of the LPWAN project.

The highest level of the network, which contains the web services, will store and analyse

data, as previously mentioned. These services are also responsible for sending alerts about fire outbursts, as well as providing all of the contents to users.

The middle level, which is composed of the coordinator, will manage all the network traffic that comes from edge devices, and is also responsible for rerouting all the received packets of information to whichever level they are destined to.

Lastly, at the lowest level of the network are the sensor nodes, which will be the focus of this dissertation. Therefore, it is the aim of this dissertation to develop the architecture of the sensor nodes for this LPWA network. These devices should be equipped with sensors that allow fire detection, and send all the data to the upper levels by using LoRa technology.

By using the wildfire prevention and detection needs as a case study, the development of the sensor nodes' architecture will base itself on its insertion in a forest with the objectives of monitoring in order to detect and send information about possible fire occurrences.

1.3 Objectives

After taking into consideration the motivation, it is the aim of this dissertation to develop of a sensor node. As so, the main goals of this dissertation are as follows:

- Study of the different Low Power Wide Area (LPWA) technologies;
- Development of a node sensor architecture that communicates with the upper network layers through the use of a coordinator;
- The use of LoRa for long range communications;
- Study of low cost and low power consumption sensors that allow to directly or indirectly detect fire occurrences;
- The nodes should be battery operated;
- Enclose all the electronics to protect it against environmental hazards.

1.4 Dissertation Structure

This document is split in six chapters, and its structure follows a logical order according to the development process that occurred during this *Master's Thesis*.

The first chapter introduces the current technological concepts, referring the context and the motivation for the development of this project, as well as its objectives.

The second chapter explores the concepts which are the basis of this project, and thus it gives a more in-depth overview of WSNs, CPSs, and LPWANs. It has emphasis on the LoRa technology, as it was within the goals of this project to use this technology. It is also mentioned in this chapter the State of the Art for the different kinds of sensors that exist and are considered for use in this project.

The third chapter gives an overview of the system, and a further selection of which components were chosen and the reasoning for their choosing.

The fourth chapter is divided into two sections, corresponding to the hardware and software implementations. It focuses on how this project was developed, and explains the path taken.

Chapter five shows the tests that were made, along with some considerations about the obtained results.

Chapter six presents the main conclusions relative to this project, as well as future improvements that can be made.

Chapter 2

State of the Art

In order to develop a sensor node for a Low Power Wide Area Network, there are some technological concepts that need to be understood. It is important to fully understand the relevance of Wireless Sensor Networks, Cyber Physical Systems, and the different existing LPWA technologies.

After knowing what a typical sensor node is composed of, and its architecture, it is then possible to make further studies on what type of components it can have.

It is also necessary to study the different types of sensors that are relevant for fire detection, as well as sensors for weather forecasting.

This chapter presents a technological overview and discussion on the topics previously mentioned, which highlights advantages, disadvantages, and relevance for this dissertation.

2.1 Wireless Sensor Networks

2.1.1 Introduction

Military actions have always been a major cause for technological development. The need for surveillance in conflict zones led researchers to create Distributed Sensor Networks (DSNs) [11] within the United States Defense Advanced Research Projects Agency (DARPA). They aimed to have several low cost sensing nodes, whose purpose was to operate autonomously, while still directing information through each other. Although the concept was the first towards creating sensor networks, there were issues that prevented its success, such as the big sensor size and the lack of wireless connectivity.

The advances in computer and networking technologies in the 1990s allowed Wireless Sensor Networks (WSNs) to fully develop. New sensors, with smaller size and lower price, were important

factors in these changes. Being the main contributor for this research, DARPA launched a program called Sensor Information Technology (SensIT) [12], which was the breakthrough for this area. This program was military based and aimed to develop networks that could retrieve sensor data, process it and display all the information in an easy to read manner, in order to provide accurate information to soldiers on the battlefield.

WSNs are composed of several independent devices (embedded systems), also known as nodes, end devices, or edge devices, that collect data from physical variables and the environment by using sensors [13]. These devices make it possible to gather information easily, and further agglomerate all the data in one place.

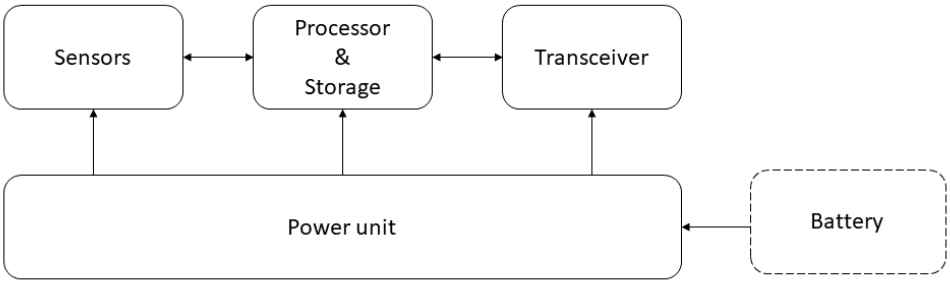


Figure 2.1: Typical node hardware architecture. (Adapted from [14])

Typically, a node is composed of four parts: the power unit, processing unit, sensing unit, and a transceiver [14].

The sensing unit is composed of all the sensors that are part of the node, which are specific to each application. They collect information on the environment and provide the output relative to each variable it is measuring. The hardware interfaces of the sensors should be chosen carefully, as there can be some external components that might be required to transform the signals into an output that is readable by the processing unit.

The power unit is responsible for providing power for all the circuitry. As these devices might be deployed in areas where electricity is not available, they are often powered by battery. This brings some power requirement metrics that should be taken into account when choosing and developing a node's hardware, and possibly the addition of energy harvesting methods.

The processing unit processes the data that comes from the sensors, and will act accordingly, whether it is by sending the information to upper levels, or by actuating on the environment. These

processors should have low processing capabilities, as they will not perform complex functionalities, and should be oriented for low power operation. Typically, the chosen processing unit should take into account the available budget, and therefore this choice should be carefully considered as to fulfill the system's specifications while not being too expensive.

The transceiver unit is what makes the interface with the upper levels by using a radio module. Generally, this unit includes an antenna, which can either be internal (PCB assembled) or external.

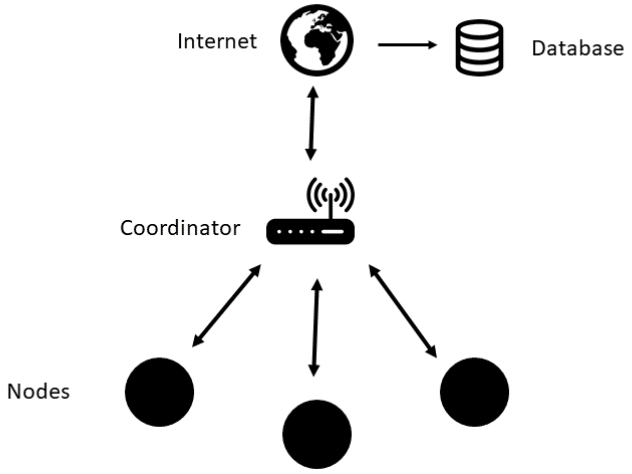


Figure 2.2: Typical WSN structure.

A typical WSN structure, as shown in Figure 2.2, is composed of a central station (base station), also known as the coordinator, which deals with all the data that comes from the sensor nodes. These coordinators might have a connection to the Internet or a web service, which allows the information to be stored and reach the user [2].

There are several ways which the nodes and the coordinator can be connected, which basically define the network topology.

2.1.2 Network Topologies

Networks can follow different patterns of information flow. As so, there have been distinct topologies for networks that can be used by WSNs.

2.1.2.1 Star Topology

In this topology, all nodes are connected to a central device (base station or gateway), as seen in figure 2.3. Information flows between each node and the central station back and forth. The nodes do not communicate with eachother.

Although this type of network is very simple, it is required that the nodes be in reach of the central device.

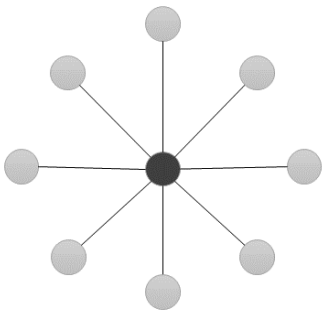


Figure 2.3: Star Topology. The lighter colour represents the nodes, while the darker colour represents the central station.

2.1.2.2 Mesh Topology

This topology is based on the nodes communicating with other nodes that are in range. This can be used to send information to any other node, or the coordinator. Mesh network topology can be seen in figure 2.4.

One of this topology's main advantage is that, if the node that it is intended to communicate with is not in reach, the message can flow through other nodes until it reaches its destination. This means that not only will it have redundancy, but also it will be more scalable, as introducing nodes should be fairly easy. However, such a vast number of communications makes this topology more prone to higher energy consumption on the nodes.

2.1.2.3 Other Topologies

There are other network topologies that can be used, such as the tree topology, or other more complex topologies, or even mixes of several different ones.

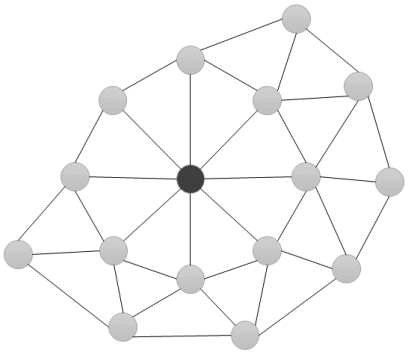


Figure 2.4: Mesh Topology. The lighter colour represents the nodes, while the darker colour represents the central station.

2.2 Cyber-Physical Systems

Cyber Physical Systems (CPSs) are systems that contain physical and computational components [15]. Through a combination of both of these components, it is possible to collect data and control physical variables through the use of sensors, while having small devices with processing capabilities interconnected in a network. By monitoring and collecting information on the physical variables and providing processing power to devices, it is possible to monitor physical processes and further actuate, inducing changes in the environment where the CPS is deployed.

CPS development has progressed due to the proliferation of sensors and actuators, along with the development of low cost computational devices with low power consumption, and the development of wireless communication technologies[16].

Taking into account the division on Cyber Physical Systems into the two components, it is important to mention that the physical part of the system is controlled by the computational part, which, apart from the processing functions, also has connectivity.

It is precisely due to having connectivity that CPSs are different from conventional embedded systems, as they are more complex, configurable, and scalable [17]. This allows them to be connected to each other, creating a system that is composed of several sub-systems, in which each device is a sub-system itself. The connection between devices can be done by either physical or wireless connections [18], and the flux of information between all systems must be managed, while also allowing more efficient actuation responses.

The interactions between devices bring a higher level of complexity and also raise the amount of possible vulnerabilities of the system. It is then crucial to take into account security factors

in order to minimize the number of susceptibilities. Such security requisites to keep in mind are related to confidentiality, integrity, availability and authenticity.

As a whole, Cyber Physical Systems enhance Wireless Sensor Networks, adding features and complexity, as well as actuation properties on the environment.

2.3 Low Power Wide Area Networks

Systems made for wider areas imply the use of longer range communication technologies. Taking this into consideration, and also keeping in mind the low power requisites for IoT systems, the development of LPWANs was crucial for these systems. These networks are not restricted to specific implementations or protocols, and is comprised of both proprietary and open-source networks [19]. Their main focus, as the name itself suggests, are battery operated devices, in order to make the most out of their lifetime, in long lengths of area.

By combining the use of sub-GHz frequencies and low data rates, LPWANs can cover extensive areas, which are estimated to be between 5 and 50 km, allowing low consumption devices to connect to each other. Conventional wireless networks (such as Wi-Fi, Bluetooth, ZigBee etc) have shorter ranges, as well as a higher power consumption for machine-to-machine communications [3], when in comparison to LPWANs.

Although cellular networks have a vast coverage, they have constraints for power consumption, as they they do not allow the energy efficiency needed for devices to keep running for several months, or even years. These types of networks also have waveforms that are more optimized for voice, messaging and high speed data, thus raising the complexity and costs of setting up a monitoring network. Therefore, they are not adequate for systems that require low complexity and low power consumption.

Figure 2.5 illustrates the differences mentioned above between cellular networks, ZigBee, and LPWA technologies. Some of these properties are related to each other, such as the number of base stations needed being directly related to the range and coverage. This means that if the range, and therefore geographical coverage, is smaller, then more base stations will be required to cover as much area as another technology with longer range would. Also, the bigger the bandwidth, the higher the power consumption will be.

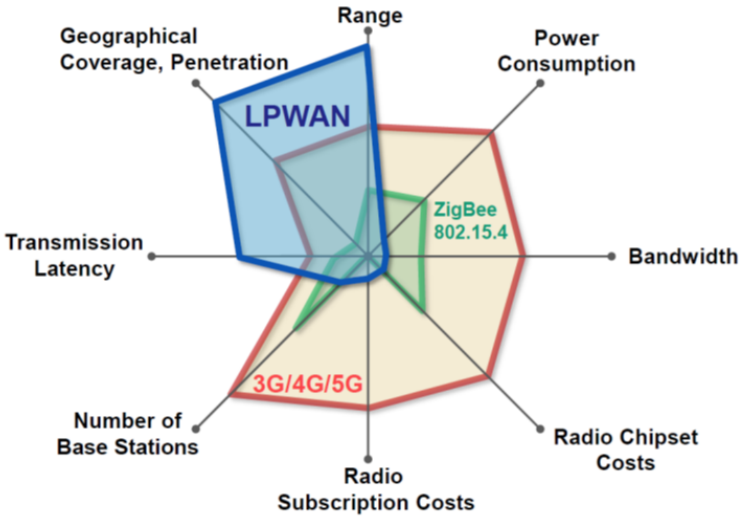


Figure 2.5: Comparison between LPWANs and other wireless technologies (Source: [20])

As so, for applications with sensor networks that require a long range and that use battery operated devices, LPWANs become a desirable solution, even though with limitations imposed by low data transmission rates. This means that LPWANs should only be considered for applications that don't require high data transmission rates and that aren't affected by delays.

Non-critical systems don't have high reliability requirements, such as sensors [21], and usually have several devices connected to the network, which have low amounts of data, making energy efficiency a priority rather than performance. Hence, an assumption can be made that LPWANs are ideal for non-critical IoT systems.

In short, LPWANs enable many non-existing IoT solutions, while also making them easier and cheaper, although introducing a natural trade-off between cost and security.

There are several types of LPWA technologies. From these, it is worth to mention SIGFOX, LoRa, RPMA, and NB-IoT, which will be further explained in detail.

2.3.1 SIGFOX

SIGFOX is a proprietary LPWA wireless network that uses Ultra Narrow Band modulation. By using this type of modulation rather than wider signal bands, SIGFOX can make a better use of channels, thus giving the network more capacity, while reducing interferences. SIGFOX operates in the unlicensed open frequencies of 902 MHz in the United States of America, and 868 MHz in Europe.

For radio transmission, Binary Phase-Shifting Key (BPSK) is used [22], which is usually used in technologies such as RFID or Bluetooth. This process consists of a phase modulation system that uses two phases for data signalling, by turning 0's and 1's in a binary message into two distinct phase states (0° or 180°).

SIGFOX is composed by several base stations installed in different locations [23]. As a collaborative network, it allows the addition of a new base station without having to reconfigure the whole network. This is due to the base stations not detecting one another, but rather just further extend the network. However, configuration and maintenance are restricted to the company, which means that it can't be remotely managed.

This brings some restrictions to SIGFOX, as it started being based on exclusivity contracts for its usage with communications companies. This implies that, for a certain area, there cannot be two different SIGFOX networks. Consequentially, it means that some sort of subscription will have to be made to a network operator that controls a certain area.

Uplink message size is small, with only 12 bytes. Along with the protocol payload, the total data size to transport is 26 bytes. Although it is a reduced size, and not much information can be sent, it reduces energy consumption for transmissions. Per day, only a maximum of 140 uplink transmissions can be made [24], and a the wireless throughput can be up to 100 bits per second.

However, SIGFOX does not have confirmation for messages, due to restrictions for networks that operate in unlicensed frequencies. As a consequence, downlink messages are restricted to 4 messages per day, with 8 bytes each. Therefore, it is likely that some messages won't reach the base stations, which cannot be detected by the nodes. In order to raise the probability of messages reaching their destination, redundancy is used, and thus a certain message is sent several times through different channels. As an example, one message can be transmitted 3 times on 3 different channels. This will have an impact on energy consumption, as sending more frequently will imply as many times more energy consumption as the number of times a message is transmitted.

2.3.2 LoRa

Long Range (LoRa) is a proprietary technology developed by Semtech Corporation that implements a physical layer for a Low Power Wide Area Network. It is highly oriented for long range and low

power consumption, while also providing secure bidirectional data transmission and the use of encryption methods [25].

LoRa uses a patented modulation technique that is based on chirp spread spectrum (CSS), and operates on the Sub-1 GHz band. The 2.4 GHz band is the conventional frequency used for wireless signals, as it is regulated worldwide. However, LoRa avoids this band due to it being more prone to interferences and its poor propagation.

By using its own modulation, LoRa is able to provide long range and a reliable signal, while taking into account energy metrics. These networks also allow thousands of devices (nodes) to connect to them.

Unlike SIGFOX, LoRa allows a network to be deployed by anyone [22], thus being less restricted. Therefore, in order to deploy a network, it is only required that the chosen hardware and gateways follow the LoRa specifications. This way, it is possible to created public or private LoRa flexible networks.

However, up to this date, the only company that produces and commercializes radio transceivers for LoRa is Semtech. This results in a market controlled by this company, and the lack of other silicon manufacturers for this technology prevents competitive prices for radio modules.

On top of the Long Range (LoRa) modulation, protocols for MAC layer can be implemented to define interoperability between the physical layer and the rest of the network.

2.3.2.1 LoRaWAN

Over the physical setup, LoRa Alliance has developed a protocol that implements the MAC layer and specifies the communications and the network architecture [26].

Operating in the frequency bands of 433 MHz and 868 MHz in Europe, 915 MHz in North America, and 430 MHz in Asia, this specification defines the network deployment as a star-of-stars topology. This means that, for this protocol, the system is composed by end nodes, gateways, and a central system. The gateways are used to transmit the messages between the nodes and a central server (Network Server) through a backhaul [27]. The Network Server is then responsible for interfacing the data coming the LoRaWAN network, and pass it to the Application Server, which is developed by the user. Usually, this is done through an already available Network Server, such as The Things Network (TTN) [28]. The LoRaWAN architecture is represented in Figure 2.6, and

its topology can be defined as a star-of-stars due to the existence of several gateways that are linked to many devices.

None of devices is tied to a specific gateway, thus the data sent can be received by several gateways, as the area of their coverage could overlap. Each gateway will forward the messages to the cloud server.

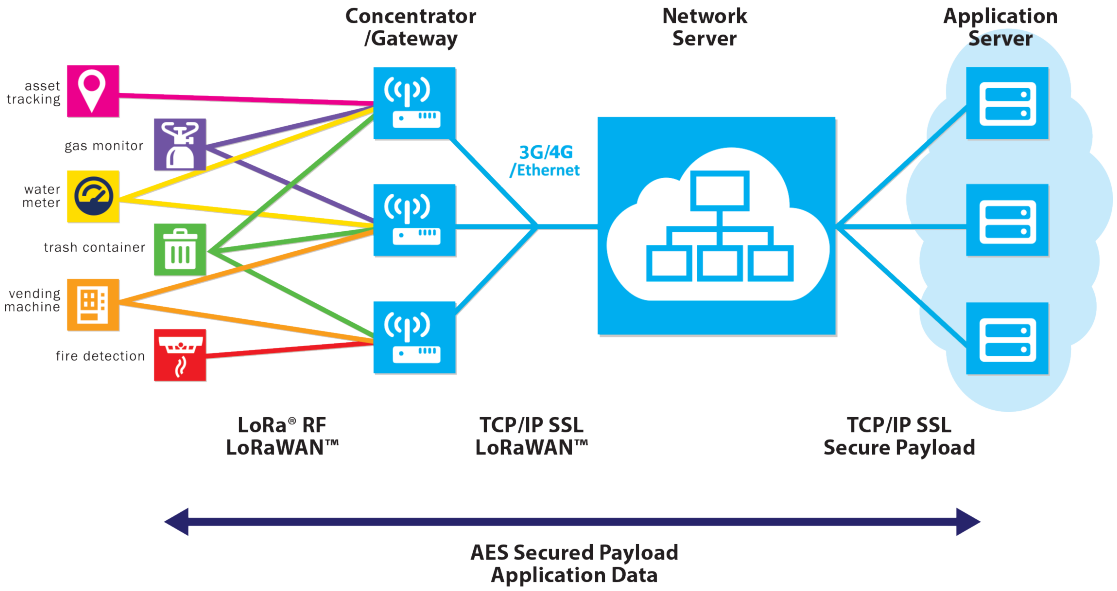


Figure 2.6: LoRaWAN network topology. (Source: [29])

While nodes (represented on the left side of figure 2.6) connect to the gateways through the use of LoRa technology, the gateways connect to the server with Internet Protocol (IP) connections, whether by Wi-Fi, Ethernet, cellular data, or satellite. The server needs to filter duplicated messages sent by different gateways, by checking their consistency and, when needed, sending acknowledge messages (ACKs). By getting duplicated messages, it is possible to make a selection of the ones that present higher quality, but also enables the capability to triangulate the node's position, in a method similar to GPS. This can be done through calculations using the time elapsed from when the messages reached the different gateways.

In order to keep the network from being compromised, LoRaWAN implements some security measures to prevent tampering. By relying on AES-128 encryption for messages, all the way from nodes to the application server. Each device is also assigned a unique IEEE EUI identifier.

Communication between edge devices and gateways is made using different channels and different data rates [30]. The selection of the transmission rate, which can go from 0.3 kbps up to 50 kbps [31], depends on the distance and the message length. To maximize energy efficiency,

LoRaWAN uses a method called Adaptive Data Rate (ADR), which controls the data that is sent by adjusting the transmission rates according to the distance from the gateway in order to get the most out of the device's location [32].

As these networks are meant to support thousands of devices with distinct functions and requirements, LoRaWAN specifies a division into three different classes of devices [33]. This division, along with LoRaWAN's place in the layers division, can be seen in Figure 2.7, where the bottom represents the physical layers with LoRa's modulation and regional ISM frequency bands, and the upper level are represented on the upper parts of the figure, respectively. LoRaWAN, as mentioned before, is implemented in the MAC layer, and therefore that's where the classes are inserted. The three classes are shown in the figure, and a brief description of each is given below.

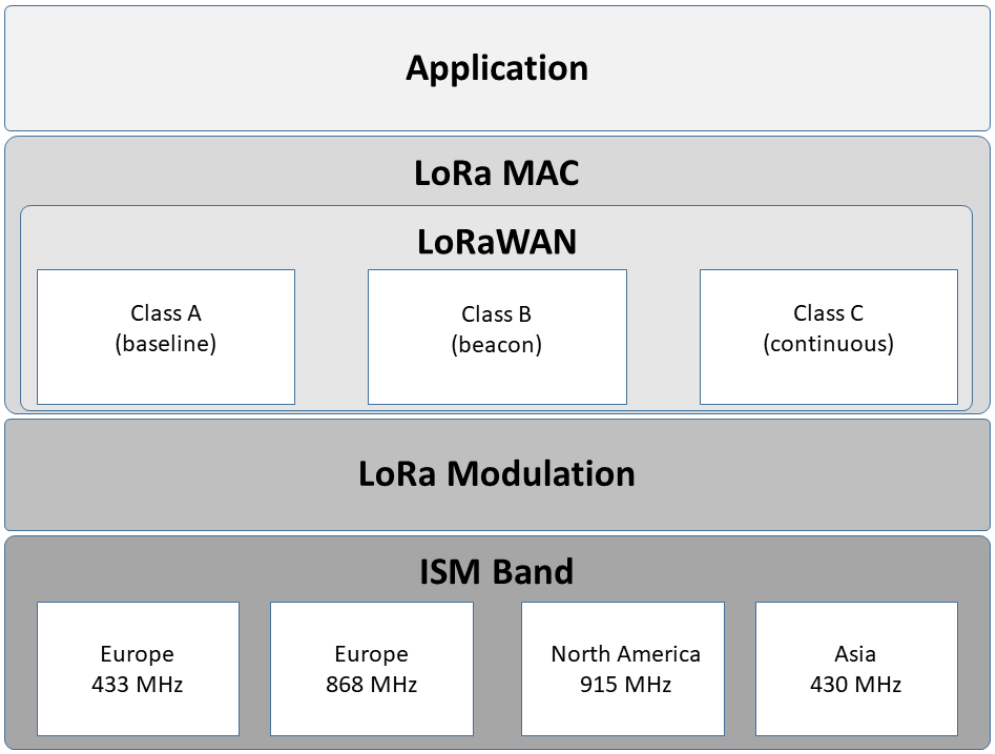


Figure 2.7: LoRa and LoRaWAN architecture. (Adapted from [33])

All the three classes have trade-offs relative to downlink, power consumption, and latency.

Class A (class for all) is the basis of the LoRaWAN protocol, and must be supported by all devices. It is meant for battery operated devices (typically with sensors or actuators with no latency constraints), and thus being the lowest power class. It allows bidirectional transmissions. For each transmission initiated by the node, there are two predefined received windows, in order to assure that a reply can be received from the server. This class is aimed at systems with

devices that have energy efficiency requirements, and therefore the communications are short and initiated by the devices themselves. Any transmission from the server to the devices can only happen after the device started the communication, and within the set receive windows.

Class B (beaconing class) is aimed at devices with low latency, but have the need to have extra reception periods. It brings the addition of fixed time and server initiated downlink slots. This class is a low-power option, however its consumption will be higher than in class A, especially depending on the latency of the application. Class B devices will open several receive windows at scheduled times. In order to know when to receive, the devices will get a time synchronized beacon from the gateway, so the server knows when the end node is listening.

Class C (continuously listening class) is meant for devices that are constantly receiving data. They will consume the most, but will offer the lowest latency. They have almost continuous receive windows that will only close when the device is transmitting. Devices in this class implement the same method of two receive windows as class A devices, however, they will not close the second receive window until they have to send data. It is up to the application server to manage the class C devices, based on the contract passed during the join procedure.

These classes allow LoRaWAN to be more flexible and scalable when it comes to the possibility of different devices to be able to connect to the network, thus providing a wider range of applications that can be developed using this technology.

2.3.2.2 Symphony Link

Symphony Link is a proprietary protocol developed by Link-Labs to operate over the LoRa physical layer, similarly to LoRaWAN. This protocol ensures that every message has a response, or acknowledgement, in both uplink and downlink messages, thus providing insight on whether the message reached the destination or not. One of its strongest advantages is the ability to update devices' firmware over the air, which helps reducing the need to have manned operations on each device to update it. Symphony Link also works on both licensed and unlicensed frequency bands, which enables the removal of duty cycle limits. Like LoRaWAN, it implements Adaptive Data Rate techniques. However, since it is a proprietary protocol, there is dependency on the company when it comes to gateways and the usage itself, which have to be acquired through Link Labs, thus making it less flexible.

2.3.3 Random Phase Multiple Access

Another LPWA technology is Random Phase Multiple Access (RPMA) from Ingenu, which uses unlicensed 2.4GHz band [34]. As these frequencies have less regulations, it allows a more uniform implementation on a global scale. Although up to this date RPMA is only available in certain locations within the United States of America [35], it has proven itself efficient in providing coverage even on remote areas with obstacles. This has allowed its use on several industries such as petrol and gas, whose infrastructures contain a lot of obstacles. Due to the facilities locations of these industries being located in remote areas, RPMA is efficient in providing the necessary coverage required. Compared to other LPWA technologies, RPMA provides a wider range, and less costs for its base stations. However, sensor nodes are typically more pricey than its competitors.

Transmission rates for RPMA are around 32 kb/s for download, and 15.6 kb/s for upload. Networks that use this technology can either be public of private, and offer AES 128 encryption for messages. Confirmation messages are also available in order to assure that there are no issues in data transmission.

Due to the lack of global testing, battery lifetime is not yet optimized for devices in this type of network, in order to have a consistent and long duration, as it is expected in this type of networks.

2.3.4 Narrow Band IoT

Narrow Band IoT (NB-IoT) is another LPWA technology that allows the coexistence with cellular networks (2G, 3G, and 4G) [36] with just some minor upgrades over the Long Term Evolution (LTE) infrastructure and operates in the licensed spectrum. Being developed by 3GPP since 2016, NB-IoT has as a main goals the low power consumption of devices, reducing costs and signal complexity, while providing a wide coverage. As it operates over the LTE infrastructure, it uses the same frequencies, although in a narrower band (180 kHz) [37]. Focusing on reducing transmission rates and bandwidth allows reducing cost, as well as power consumption. NB-IoT uses using Quadrature Phase Shift Keying (QPSK), which is a system that modulates digital signals onto a radio-frequency signal that uses four phase states to code two digital bits. This allows more efficiency in the bandwidth, by improving performance and throughput even in the presence of interferences.

Messages are sent with transmission rates that can go up to 250 kbps, and they're mostly asynchronous. Security and privacy are two important aspects of NB-IoT, as it implements some of LTE's mobile networks features that support user identity, entity authentication, data integrity and mobile device identification.

Due to delays in being commercialized, NB-IoT still does not have sufficient data on real applications, and no information about its impact on battery lifetime in devices is available.

2.4 LoRa and LoRaWAN

Since it was intended to use LoRa, along with LoRaWAN, for the development of this *Master's thesis*, a more in depth explanation of the protocol is provided. Even though there are directives for how the Application Server is to be implemented, focus will be given on the end device specifications, as it is within the scope of this dissertation. For the Network Server, it was considered TTN, which can work as both Network Server and Application Server.

It is important to notice that, even though the current version of the protocol is LoRaWAN 1.1, at the start of this dissertation, and throughout most of its development, The Things Network had only supported LoRaWAN 1.0. In consequence, the sessions will be explained for both versions, although the rest of the specification and the dissertation implementation infers to LoRaWAN 1.0. This is done with the intent to mention the complexity changes that were introduced in the newest version of the protocol, mainly in the number of required parameters and security handling.

In section 2.3.2.1 an overview of LoRaWAN and its the different device classes is given, along with their functions. For the purpose of this project, class A will be used, as it is intended to have the lowest power possible, and this is the class that satisfies this requirement. However, it is important to notice that using this class brings the limitation of not allowing for a device to get anything from the server except on the specific receive windows. For the purpose of this work, this is not taken as a major issue, as the LoRa module should not be working unless when transmissions are made, due to power saving goals.

LoRaWAN is restricted by the regional regulations for ISM bands. In Portugal, operation can be done within one of the following frequency bands: 433.05 MHz to 434.79 MHz (designated as EU433), or 863 MHz to 870 MHz (designated by EU863-870 or EU868).

To communicate with gateways, the nodes can use a set of different channels within the specified frequencies.

Some important concepts to know about LoRaWAN are its definitions of the different components of the network. These are explained below.

2.4.1 Sessions

LoRaWAN works in sessions, where devices have a session with the Server. Sessions are divided into Network Sessions, and Application Sessions. The Network Session is maintained by the Network Server and the node, whereas the Application Session is maintained by the Application Server and the node.

The Network Server handles the messages received, while the Application Server is responsible for the data that is sent in the message payload, which in this case refers to the node information gathered from the sensors.

In order to be a part of a LoRaWAN network, there is a set of data contents that need to be stored in the device beforehand, and that are required for devices' activation on a network. After the activation process, a new security session context is established between both the servers and the device.

There are two ways of activating a device on the network: activation by personalization, and over the air activation.

It is important to note that, regardless of the activation process that is used, in neither should it be able to derive keys from acquired information, such as stolen keys.

2.4.1.1 Over-The-Air Activation (OTAA)

For this type of activation, nodes have to go through a join procedure in order to be able to communicate with the Network Server. A new join must be done if the session context information is lost.

The node must have a set of information in order to be able to do a join procedure.

In **LoRaWAN 1.0**, The required parameters for OTAA are **DevEUI**, **AppEUI**, and a root key **AppKey**.

- **DevEUI** is the global ID in EUI address that identifies a device or node.

- **AppEUI** is the application identifier, given in EUI address, that identifies the application provider of the node. This means that if a node is to be used in a different application, it requires a new AppEUI.

- **AppKey** is an AES-128 application key. It is given by the application provider to the nodes. When an OTAA join procedure occurs, this key is used to derive two sessions keys: **NwkSKey** and **AppSKey**, which are specific to each device.

In **LoRaWAN 1.1**, the required parameters for OTAA are **DevEUI**, **JoinEUI** (instead of AppEUI), and two root keys **NwkKey**, and **AppKey**.

- **JoinEUI** is the global ID in EUI address that identifies the join server that will process the join procedure and further session keys derivation.

- **NwkKey** is an AES-128 root key. Along with the AppKey, it is given by the application provider, and it is used for the join procedure in OTAA. It is used to derive the **FNwkSIntKey**, **SNwkSIntKey**, and **NwkSEnchKey** network session keys.

- **AppKey** is similar to LoRaWAN 1.0 AppKey, except it derives one key instead of two, the AppSKey.

When joining a network using OTAA, a new session is started, and new information will be sent to the node and will have to be stored in the device. All the new given parameters are unique for each node, and are given when the join request is accepted through the join.

For **LoRaWAN 1.0**, it is given a **DevAddr**, **NwkSKey**, and **AppSKey**.

- **DevAddr** is the 32-bit device address, which identifies a device within the network it is currently at.

- **NwkSKey** is the network session key given to the device by the network. It is used for data integrity checks, through the calculation and verification of the Message Integrity Check (MIC), and for encryption and decryption of the payload of MAC data messages.

- **AppSKey** is the application session key given to the device by the network. It is used for encryption and decryption of the payload of application messages. Likewise, it is used for calculation and verification of the MIC of application data messages. This key is not known

to the Network Server, thus protecting the data that is destined to the Application Server, which can only be deciphered in that layer.

For **LoRaWAN 1.1**, it is given a **DevAddr**, the network session keys **FNwkSIntKey**, **SNwkSIntKey**, **NwkSEncKey**, and the application session key **AppSKey**.

- **FNwkSIntKey** is the forwarding network session integrity key, that is used for data integrity assurance, which is used to calculate the MIC of the uplink data messages.
- **SNwkSIntKey** is the serving network session integrity key, which is used to check the MIC of the downlink data messages, and together with the FNwkSIntKey, calculate the uplink MIC.
- **NwkSEncKey** is the network session encryption key, which is used to encrypt and decrypt commands transmitted as payload in specific fields.

When a device requests a session with the network, new keys are generated and given to the device, thus a session should be kept active for as long as possible, as the network will assume a new device within a new session request.

All the given keys should be safely stored, as they are specific to each node, and are used to authenticate into the network. However, if these keys are extracted, only the node will be compromised, and not the whole network, as a new session is assumed to be for a new device. This is one of the main reasons why Over-The-Air Activation is the preferred method, and because it makes the deployment process simpler for large quantities of devices as keys are generated on request.

2.4.1.2 Activation By Personalization (ABP)

This method skips the join procedure by having the device address and the session keys stored in the device (hardcoded) and not derived from other keys, and thus each one is unique to a given node and will always be the same throughout any given session.

The node will then have the required information for being a part of a network stored in it before connecting itself. For LoRaWAN 1.0, these parameters are the device address **DevAddr**, and the two session keys **NwkSKey** and **AppSKey**. For LoRaWAN 1.1, these parameters are the device

address **DevAddr**, and the four session keys **FNwkSIntKey**, **SNwkSIntKey**, **NwkSEncKey**, and **AppSKey**.

If the device is tampered with, and the keys are taken, it is possible to recreate the node, and inject malicious information into the network, thus OTAA is preferred due to having higher security. This is due to OTAA creating a new session every time a join procedure is done, making a new session be assumed as a new device, whereas ABP works as if there is one session for each set of registered keys within the network.

2.4.2 Message Format

LoRa messages follow a specific format, which is different for both uplink and downlink messages. Uplink messages are sent from the nodes to the Application Server, going through one or more gateways and the Network Server in the process, as previously explained. Downlink messages are sent from the Application Server to the node, by following the opposite path.

Since both uplink and downlink messages follow a different and specific format, it is important to know how LoRa implements its message structure, in order to be able to implement LoRa communications.

All LoRa packets, regardless of whether they are uplink or downlink, contain the following structure: a preamble, an optional header, and the data payload.

Data messages are divided into two types: explicit mode and implicit mode. By default, explicit mode is used.

Preamble

The Preamble, which is at the beginning of the messages, contains important information for the receiver regarding the data transmission, such as the synchronization word. Its size is variable, and can be changed through a register, although an extensive Preamble can cause overhead. By default, it's a 12 symbol sequence, although its length can vary from 10 symbols, to 65539 symbols [38]. The receiver should expect a Preamble size that is not smaller than the actual size of the Preamble that is transmitted.

Header

The header, which is optional depending whether the message is in explicit or implicit modes, provides the payload size, in bytes, the code rate, and an optional payload Cyclic Redundancy Check (CRC) extension. This header has its own CRC, in order to avoid errors.

Data Payload

The payload, also known as PHYPayload, contains the data, and will be further explained in the following sections.

2.4.2.1 Join Procedure

The session is initiated by the device, which starts by activating itself in the network through a join procedure. This is done by exchanging messages with the server. As previously mentioned, this process only occurs for Over-The-Air Activation.

The first message sent by the node is a *join-request*. This message is not encrypted and contains a MAC message which is composed of the App EUI, the DevEUI, and a DevNonce (as seen in figure 2.8), which is a random value. The DevNonce values, which are kept in the network server, serves as a way to avoid attacks that try to infiltrate the network by mimicking the join process of a node, as join requests with repeated values of this parameter are ignored.

Figure 2.8: *Join-request* message format. (Adapted from: [39])

The *join-request* is followed by a *join-accept* message, which provides the device with the necessary values for it to derive the NwkSKey and AppSKey, as well as its DevAddr. The sessions keys are not explicitly given for security purposes, as they are critical for a device's authentication within the network. Response is only given if the node is accepted in the network, otherwise, no response is given.

2.4.2.2 Data Messages

The default message structure for uplink (explicit mode), represented in figure 2.9) is composed of a Preamble, a Physical Header (PHDR) and its header CRC (PHDR_CRC), and the Physical

Payload (PHYPayload) and its own CRC (Payload CRC). The PHYPayload will be explained in the following section. Both CRC fields are intended to assure the integrity of the data.

The physical header and CRCs are done by the radio transceiver.

Uplink PHY				
Preamble	PHDR	PHDR_CRC	PHYPayload	Payload CRC

Figure 2.9: Explicit mode uplink message format. (Adapted from: [39])

For messages with known payload, coding rate, and CRC contents or size, the header can be omitted, in order to reduce the message size and thus the transmission time. The parameters specified in the header should be manually configured on both sides of the radio transmission, as they are not included in the message.

This is called the implicit mode. This way, the uplink data message will only be composed of the Preamble, the PHYPayload, and its respective CRC, as seen in figure 2.10.

Uplink PHY		
Preamble	PHYPayload	CRC

Figure 2.10: Implicit mode uplink message format. (Adapted from: [39])

The downlink messages sent from the server to the nodes are similar to the uplink messages, with the exception that they don't have Payload CRC. This is due to the duty-cycles restrictions, so it is intended to keep the messages as short as possible.

The *join-accept* messages are treated as a normal downlink message.

Downlink PHY			
Preamble	PHDR	PHDR_CRC	PHYPayload

Figure 2.11: Downlink message format. (Adapted from: [39])

2.4.2.3 Physical Payload (PHYPayload)

The Physical Payload (PHYPayload) is composed of a MAC header (MHDR) with the message type and the version of the format that is used for encoding, a MAC Payload, and the Message Integrity Check (MIC). This structure can be seen in figure 2.12.

The message type (MType) defines whether the message is a *join-request*, *join-accept*, or confirmed or unconfirmed uplink or downlink. There are a total of six different message types.

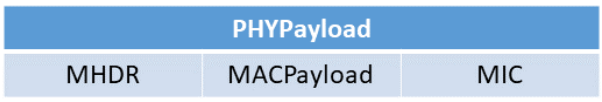


Figure 2.12: Physical Payload structure. (Adapted from: [39])

While an unconfirmed message will not have any reply, a confirmed message must always be acknowledged by the receiver, which will be further explained in detail in the following sections.

2.4.2.4 MAC Payload

The MAC Payload is what contains the data, and it is composed of a frame header (FHDR), an optional port field (FPort), and an optional frame payload field (FRMPayload), which is the application payload. This structure can be seen in figure 2.13.

The frame payload is encrypted using one of the sessions keys. The key used for encryption is dependent on the FPort field: if this value is 0, the NwkSKey is used to encrypt the message, if FPort has any other value up to 255, the AppSKey is used instead.

After the frame payload being encrypted, the Message Integrity Check (MIC) is then calculated. The MIC is calculated for the MAC header and all the MAC Payload fields combined.

Figure 2.13: MAC Payload structure. (Adapted from: [39])

The FPort field defines whether the payload has MAC commands only, or if it has data. This field is absent if the payload is empty.

The frame header contains the device address (DevAddr) previously mentioned, the frame control (FCtrl), a frame counter (FCnt), and the frame options (FOpts). Its structure can be seen in figure 2.14.

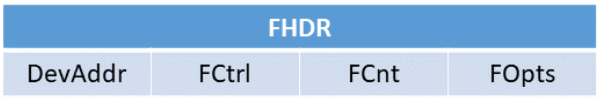


Figure 2.14: Frame header structure. (Adapted from: [39])

The FOpts field is used to store MAC commands.

The frame counter (FCnt) is divided into two frame counters that keep track of the data frames, both for uplink and downlink transmissions. These counters are kept in synchronization between

both the end device and the Network Server, and can be used to know how many data frames are lost. This value is reset after a join procedure.

The FCtrl field contains important the fields relative to the messages that are being sent, and has two variants: one for the uplink transmissions (shown in figure 2.15), and another for the downlink transmissions (shown in figure 2.16).

The common fields for both FCtrl variants are the the Adaptive Data Rate bitsClass (ADR and ADRACKReq), which allows enabling or disabling this feature, the message acknowledge bit (ACK), which is set when the receiver is responding to a confirmed data message, and the frame options field (FOptsLen). FOptsLen contains information about the size of the FOpts field, which is included in the frame header as mentioned previously. The RFU bit is reserved for class B devices, where it should be set if a device is issuing uplinks while operating as a class B device.

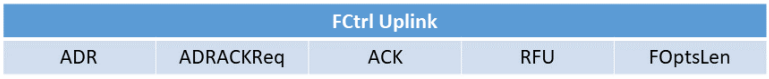


Figure 2.15: Uplink Frame Control structure. (Adapted from: [39])

On the downlink messages, there is a FPending bit, which indicates whether there are more messages to be sent, and if so the node should be able to receive more data by opening more receive windows.

Figure 2.16: Downlink Frame Control structure. (Adapted from: [39])

The MACPayload's maximum length, which can be seen in table 2.1, is region specific and dependent on the data rate settings. Its size is also given by the existence or absence of the FOpt field. For this dissertation, it will be considered the worst case scenario (smallest payload).

Table 2.1: Data Rate and respective maximum MAC payload size (in bytes) for EU863-870 band.

Data Rate	Payload
0	51
1	51

Continues on the following page

Table 2.1 – Continuation of the previous page

Data Rate	Payload
2	51
3	115
4	222
5	222
6	222
7	222

2.4.3 Message Transmission

There are some parameters that can influence the transmission of LoRa messages, such as the Bandwidth (BW), the Spreading Factor (SF), and the Data Rate (DR). Each of these parameters can be configured, although some of them are either related or dependent on others.

The signal Bandwidth is the difference between the lowest frequency of the signal, and the highest frequency of the signal. The higher the available Bandwidth, the higher the amount of data that can be transmitted (bit rate).

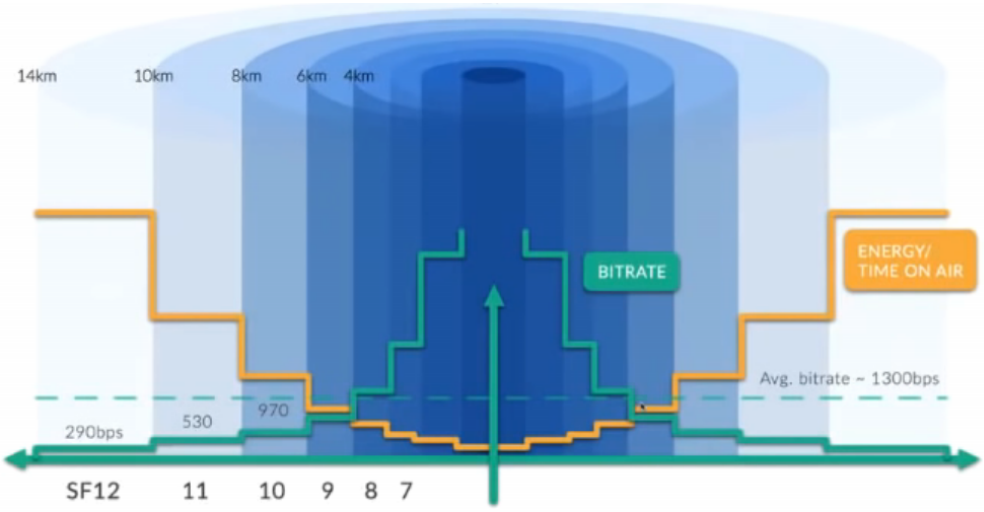


Figure 2.17: Parameters that affect LoRa message transmission. (Adapted from: [40])

LoRa data is basically sent in small pieces, that are called *chirps*. These *chirps* encode the LoRa data. The Spreading Factor indicates how fast or slow those *chirps* are, which means, how much data per second is sent. This means that the *time on air* of the messages can vary with

the Spreading Factor configuration, as seen in figure 2.17. The values of the SF parameter can range from 7 (SF7), to 12 (SF12).

The slower data is sent, the further it can get, but also the more *time on air* and energy consumption it will require. This makes the network more scalable, as more data is sent when the nodes are at a closer distance to the gateways. Therefore, the SF parameter is directly related to the DR, as it defines the bit rate of the messages. This direct relationship can be seen in table 2.2, which shows how LoRaWAN related the Data Rate and the Spreading Factor values, and the respective bit rate.

Table 2.2: Data Rate and respective data transmission configuration for EU863-870 band.

Data Rate	Configuration	Bit Rate (bit/s)
0	SF12 / 125 kHz	250
1	SF11 / 125 kHz	440
2	SF10 / 125 kHz	980
3	SF9 / 125 kHz	1760
4	SF8 / 125 kHz	3125
5	SF7 / 125 kHz	5470
6	SF7 / 250 kHz	11000
7	FSK: 50 kbps	50000

As seen in section 2.3.2.1, LoRaWAN divides devices into three different classes, each with specific functionalities and restrictions. For each application, the correct class should be chosen accordingly, taking into consideration requirements such as energy consumption, frequency of message transmission, or the need to have a response to messages. Despite all this, all devices must join the network as class A devices, switching to another class afterwards.

The message transmission timings for the different LoRaWAN classes are explained below.

2.4.3.1 Class A

Class A is meant for devices with energy consumption metrics that do not require acknowledgment to all their messages or data sent from the server frequently. After a message is sent, and

for messages that require confirmation, the node will have two receive windows, in which any message sent from the server can be taken.

The device opens two receive windows at specified times after an uplink transmission, as seen in figure 2.18. In this figure, the darker colour represents the data that can be sent from the node, and the lighter colour represents the data that can be sent from the server. If the server does not respond in either of these receive windows, the server's next opportunity to send data to the node will be after the next uplink transmission from the device. The server can respond either in the first receive window, or in the second receive window, but should not use both windows.

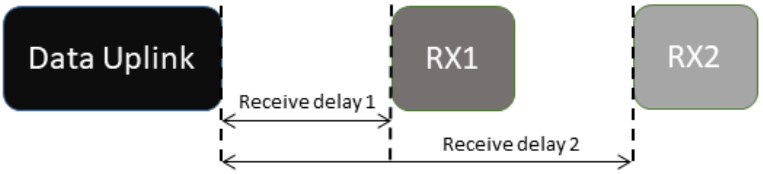


Figure 2.18: Class A message transmission. (Adapted from: [39])

The first receive window (RX1) has a duration of approximately 20 microseconds, and occurs after the node's message ends. It has the same frequency channel as the uplink message, and the data rate is similar to the last sent message, although these parameters are region specific. The second receive window's (RX2) data rate and frequency are configurable, and are also region specific.

To detect if a message is received by the node, its LoRa module only needs to detect a message Preamble during either of the receive windows. Another uplink message cannot be issued until either a message is received, or the second window has ended.

2.4.3.2 Class B

The beaconing class is based on the principle of specific intervals for receiving messages from the server.

Apart from the usual receive windows that are already implemented with class A, the node will also have a certain fixed time frame for data reception. In order to do so, the node needs to synchronize itself with the gateway, so it knows when data can be sent. This is done by the issuing of a beacon, which is sent periodically as a means to synchronize with the end devices. The time in between the issue of two beacon signals is called the Beacon Period. This enables

opening an extra receive window called *ping slot*, and it is what can be used to have downlinks sent from the server.

Figure 2.19 represents an uplink message transmission issued by the end device, while in between a beacon period. It shows the beacon (B) issuing, followed by the Ping Slot, and the respective uplink data message with its receive windows (RX1 and RX2).

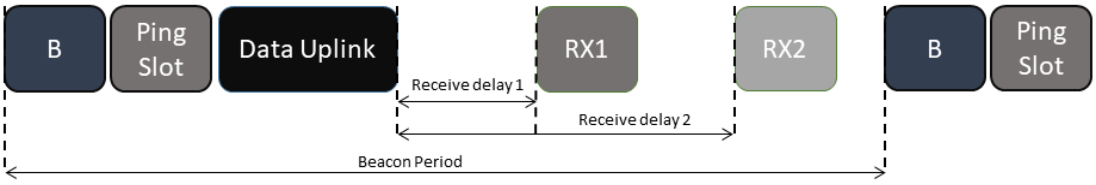


Figure 2.19: Class B message transmission. (Adapted from: [39])

When the synchronization with the server is lost, the node will go back to being a class A device, until it is able to switch back to being a class B device. The management between swapping classes is done in the software application level of the end device.

If the node changes location and a change is detected on the beacon reception, an uplink should be sent in order to adjust the downlink path to the device.

2.4.3.3 Class C

Class C is aimed for devices that do not have power restrictions. Their purpose is to always have an open receive window, so data can be sent at any time. In order to do this, the RX2 window is always kept available, except when an uplink is being issued, or the RX1 window is open (which occurs after a transmission, just like in class A).

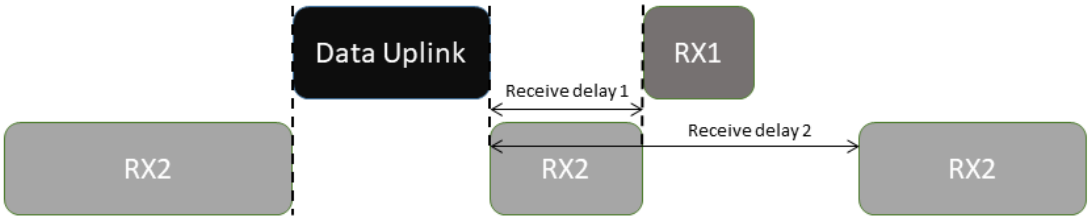


Figure 2.20: Class C message transmission. (Adapted from: [39])

The opening of the second receive window occurs with the timings associated with a transmission, but it will stay open until the next uplink is issued.

2.5 Gas Emission from Wildfires

Wildfire occurrences have a great impact not only on the landscape, but also on the air quality. Since combustion implies chemical reactions, then consequentially the results from those combustions are gases that can be either pollutants or even toxic.

As mentioned in section 1.1, pollutants such as carbon dioxide, carbon monoxide, methane and nitrous oxides are related to wildfires. Therefore, if these components are monitored and abnormal quantities are detected in the air, then it is very likely that there is a fire occurring.

As so, a few of these components were selected and some details about them will be given, as to give an in-depth perception of their characteristics, and how they are formed from fire outbursts.

2.5.1 Nitrogen Dioxide

Nitrogen dioxide (NO_2) is a toxic gas. It has a very distinct smell and a reddish-brown colour. It is a pollutant gas which has a negative impact on human health, and can cause respiratory problems [41].

Its formation happens mostly due to electrical discharges from lightning storms, as well as natural processes that occur in plants, soil, or water [41]. NO_2 is also originated from combustions, due to their high temperatures [42]. The usage of motor vehicles is one of the biggest factors that has contributed for the emission of this gas. However, there are others, such as biomass combustion, which is a consequence of forest fires.

A chemical reaction between the nitrogen atoms (N_2) and nitrogen dioxide (NO_2) occurs from high temperature combustions [43]. This reaction originates nitric oxide (NO), and is represented in equation 2.1.

$$N_2 + O_2 \longrightarrow 2\,NO \tag{2.1}$$

Equation 2.1: Nitric oxide formation equation.

On the other hand, nitric oxide reacts with oxygen, originating nitrogen dioxide. This chemical reaction is represented in equation 2.2.

$$2\,NO + O_2 \longrightarrow 2\,NO_2 \tag{2.2}$$

Equation 2.2: Nitrogen dioxide formation equation.

The presence of nitrogen dioxide in the air, especially on forest areas, can imply the occurrence of a major wildfire with high temperatures.

It exists in the air in quantities smaller than 1 ppm (parts per million).

2.5.2 Carbon Monoxide

Carbon Monoxide (CO) is toxic gas that can cause hazards in human health, eventually leading to death. It is an odourless and colourless gas that results from incomplete combustions [44].

Carbon (C) atoms react with oxygen (O_2), which are combined from the burning process, and the leading into Carbon Monoxide molecules. This reaction is represented in equation 2.3.

$$2\,C + O_2 \longrightarrow 2\,CO \tag{2.3}$$

Equation 2.3: Carbon Monoxide formation equation.

The usual concentration of carbon monoxide in the air is of approximately 0.2 ppm.

2.5.3 Carbon Dioxide

Carbon dioxide (CO_2) is an odorless and colorless gas that exists in the atmosphere. Through the photosynthesis process, plants convert carbon dioxide into oxygen [45], which is crucial for the human respiratory system.

Equation 2.4 represents the chemical reaction that occurs between carbon and oxygen that originates CO_2.

$$C + O_2 \longrightarrow CO_2 \tag{2.4}$$

Equation 2.4: Carbon dioxide formation equation.

Carbon dioxide can be formed by the body, or by combustion. Similarly to carbon monoxide, it also results from fires. Although the equations are a bit similar, the process of its formation is the opposite, as carbon dioxide derives from combustion, whereas carbon monoxide results from incomplete combustions.

The usual concentration value for CO_2 in the atmosphere is of approximately 400 ppm, although this value has been slowly raising over the years.

2.5.4 Gas Sensors

There are several methods of gas detection. Therefore, different types of sensors were created to implement such methods.

One of the most relevant aspects to take into account when studying these sensors is the coverage area in which they can detect gas. To that end, it is important to keep in mind that gas detection occurs only on the sensor itself [46], thus it will have to be exposed to the gas to be able to detect it. Several factors might influence in this process, such as wind direction, proximity to areas where gas emissions are common (cities, industrial zones, etc), or even proximity to places with high probability of fire outbursts.

Another important issue to notice are the physical properties of the gas that is intended to detect or monitor [46]. As an example, placing a sensor to detect a certain type of gas that is lighter than air and that will, evidently, ascend, should be a different setup than placing a sensor that has the opposite characteristics.

Aside from these factors, it is also necessary to study how each type of sensor works, in order to make a decision on which sensor better suits the gases that are to be detected, and thus better aimed for the scope of this dissertation.

After a study is done on the different types of sensors, it is then presented the most relevant gases that occur from wildfires, as it is this dissertation's case study.

2.5.4.1 Catalytic Sensors

This type of sensor, also know as catalytic bead sensors, converts temperature changes, which are made by chemical reactions, and transformed into electrical signals [47]. Their measurements are based on the altered resistance values given by the temperature raise, which is directly related to the concentration amount of the gas. Through the use of a catalyst to create the oxidation of a certain gas, the temperature variation is then converted into the corresponding signal by a Wheatstone bridge [48].

These sensors are composed of two beads, one which is used for the oxidation process of the gas to be measured, and another that provides a reference value, which is usually coated in glass in order to keep it from being affected by the presence of combustible gases. More specifically, these sensors use a coil, which is coated by a catalyst layer, and is heated by an

electrical current. The heat generated by the coil allows the oxidation process to happen, as it is due to the hot catalyst that the chemical reaction occurs. This process results in heat, which in turn rises the temperature of the coil and the catalyst. The electrical resistance of the coil changes with the temperature rise, and thus providing the electrical signal output. The output signal is then compared to the reference resistance value, through the Wheatstone bridge, allowing the sensor to output the level of a certain gas.

Catalytic sensors are very accurate in the detection of flammable gases, also know as combustible gases, with a fast response time. Their efficiency in gas detection makes them ideal for their use in detection systems. Moreover, they are not very susceptible to changes in humidity or temperature, as they keep their detection properties in different scenarios. However, due to the use of a catalyst, they might be subject to poisoning derived from other components in the atmosphere, or even leakage of the catalyst itself. Furthermore, these sensors are not adequate for use in the presence of several toxic gases, as they shorten their lifetime.

2.5.4.2 Electrochemical Sensors

These sensors, whose model is represented in figure 2.21, are based on an electrochemical reaction that generates an electrical current that is proportional to the concentration value of a certain gas. They are used mostly to measure the levels of toxic gases. The sensor itself is composed of a chamber that contains an electrolyte and two or more electrodes. Through a membrane, gas flows into the sensor, allowing an oxidation process to occur. This induces changes in the electrical current that are proportional to the gas concentration. By varying the materials used to make the electrodes and electrolyte, it is possible to vary the gas that is to be detected by the sensor.

Figure 2.21 represents a typical electrochemical sensor, with two electrodes for gas detection, and the reference electrode that is contained within the dielectric.

Their cost is moderate, as the values can range from 50 to 150€ [49].

These sensors are typically small, and have a low power consumption. They have a linear response, which means that the value given by its output is proportional to the amount of gas in the air. However, they can be susceptible to temperature and humidity changes, depending on which electrolyte they have. There is also the possibility of its leakage.

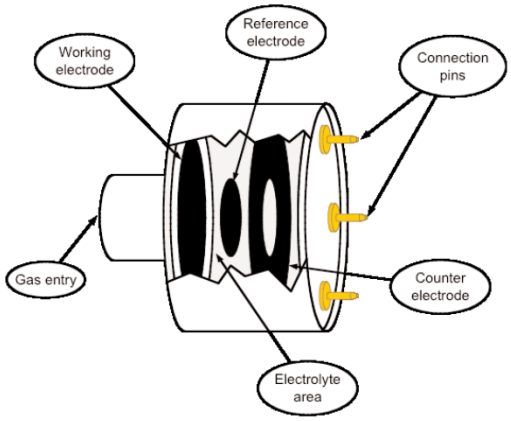


Figure 2.21: Electrochemical sensor.(Source: [48])

Some sensors require the presence of oxygen for the electrochemical reaction to happen. Moreover, some gases can also poison the sensor, giving wrong readings. Gases such as Carbon Monoxide (CO) and Carbon Dioxide (CO_2) are examples of contaminants for the electrolyte or the electrodes. Therefore, these sensors should not be exposed to high gas concentrations, as their normal behaviour can be compromised.

2.5.4.3 Semiconductor Sensors

These sensors are made of metal oxide semiconductors that vary their conductivity (or resistivity) in the presence of certain chemical elements [50]. They are able to detect gases, even in small concentrations. The resistance value depends on whether there is a reducing or oxidizing gas, affecting the resistance value of the metal.

In order for the sensor to work, it needs to be heated up to a temperature that will make the presence of the gas cause changes on the conductivity of the material. When there is no gas present, oxygen will ionize the surface and the sensor will become semi-conductive, and when there are molecules of the gas that is to be measured, they will replace the oxygen ions.

Sensor behaviour to these changes will depend on the type of semiconductor used [51]. There are two types: the n-type, and the p-type.

The p-type semiconductors conducts mostly positive charges, and thus its interaction with reducing gases causes its conductivity to decrease, as the gas reduces the concentration of positive charges. When in the presence of an oxidizing gas, the opposite effect occurs, which

means that the conductivity increases due to the increase of the positive charges provoked by the gas.

N-type semiconductors conducts with mostly negative charges, or electrons, hence an oxidizing gas with provoke a depletion of the number of electrons, consequentially decreasing its conductivity. Reducing gases will have the opposite effect, increasing the n-type semiconductor conductivity.

These variations on the resistivity of the sensors can be measured and further related to the gas concentration.

These interactions are shown in table 2.3.

Table 2.3: Types of semiconductor and respective behaviour to different gases.

Type	Oxidizing Gas	Reducing Gas
p-type	Resistance increases	Resistance decreases
n-type	Resistance decreases	Resistance increases

Metal oxide semiconductor sensors have a relatively low cost, with values around 10 to 15€ per sensor.

As they are not very specific, false positive results may occur when they detect a similar gas. They are also restricted by temperature and humidity conditions, which influence the semiconductor's resistance. However, this can be overcome through characterization and its effects can be compensated with calibration methods and adjustments to the sensors readings based on the environment conditions.

2.5.4.4 Optical Sensors

Optical sensors are based on the principle that gas molecules absorb light or other electromagnetic waves, such as infrared (IR), or ultraviolet (UV), and that this absorption occurs in a specific wavelength for a given gas. The concentration values can be obtained based on the amount of absorption that occurred, or the radiation intensity that reaches the sensor.

The intensity of absorption is given by the Beer-Lambert equation [52], which states the proportionality between the concentration of gas and the measured absorption, and is represented in equation 2.5.

$$A = log_{10}\left(\frac{I_0}{I}\right) = alc \tag{2.5}$$

Equation 2.5: Beer-Lambert equation.

In this equation, *A* represents the absorption, I_t the measured light intensity, I_0 the emitted light intensity, *a* is an absorption factor that is dependant on both the gas and the frequency, *l* is the length of the path that light has to travel, and *c* is the gas concentration.

Optical sensors are therefore based on the Beer-Lambert law.

2.5.4.5 Non-Dispersive Infrared Sensors

Visible light's measurement range for gases is reduced due to the existence of steam or smoke in gas chambers, which provokes scatter and the diffraction of visible light. Such phenomenon does not occur in Infrared light, as its wavelength is wider, making it effective on gas detection. Furthermore, the evolution of MEMS technology has allows the reduction of IR emitters and receivers, which made them more popular than other types of sensors [53].

The Non-Dispersive Infrared (NDIR) sensors, which are the most common along IR sensors, is composed of an IR emitter, an IR detector, and a gas chamber. The IR source emits to the gas chamber, where the sample of environment gas is present, and with it the gas that is to be detected. After being absorbed by target gas, the radiation goes through an optical filter that filters all the radiation except for the one in the wavelength that is absorbed by the target gas. It is then measured by an IR detector. This is further illustrated in figure 2.22.

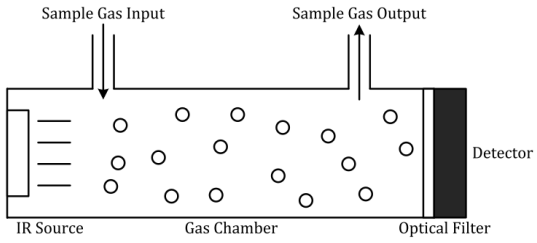


Figure 2.22: Non-Dispersive Infrared sensor.(Source: [53])

In order to improve the precision of this type of sensor, another detector can be used to remove environmental factors. While not reducing the radiation intensity that reaches the detector, a second detector can be used as a reference. To guarantee that the parameters of the IR radiation

that reach both detectors are the same, the same IR emitter is used, although with a mirror to reflect the IR waves. Figure 2.23 shows the NDIR sensor with two detectors.

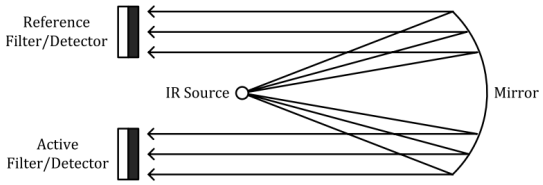


Figure 2.23: NDIR sensor with two detectors.(Source: [53])

These sensors have a high precision and high measurement range. They are also less susceptible to contamination. However, it has some restrictions to the gases that can be monitored, and the price is relatively high, with values similar to the cost of electrochemical sensors.

One of the gases that is measured more commonly by this type of sensor is Carbon Dioxide (CO_2), as these sensors have a fast measuring response and they are very precise in determining the concentration of this gas.

2.5.4.6 Photoionization Sensors

These sensors use an Ultraviolet light (UV) to ionize the target gas molecules. This process consists of removing electrons from the molecules, which creates ions. When the ions reform with electrons, an electrical current is generated, and this value can be measured to provide the concentration of a certain gas.

Gases with higher ionization potential than that given by UV lamps are undetectable by these sensors, such as methane.

Photoionization sensors have a very short response time, and are able to detect a wide number of gases even at low concentrations.

With the development of this kind of technology for gas sensing, costs have been lowering, and the price range is similar to electrochemical sensors. However, these sensors are susceptible to humidity changes, and need frequent calibration and maintenance. They are also not capable of identifying a specific gas.

2.6 Temperature sensors

Nowadays, temperature sensors are used in a vast number of situations, such as refrigerator temperature control, in cars, medicine, room temperature monitoring, etc.

There are several types of temperature sensors, which are: thermocouples, Resistance Temperature Detectors (RTDs), thermistors, and semiconductors.

2.6.1 Resistance Temperature Detector (RTD)

An RTD sensor is based on the principle that with temperature changes, metal resistance also varies [54]. This variation can be measured and further correlated to the temperature. The linearity of its output makes measurements very accurate.

RTDs are made of conducting metals, such as platinum, copper or nickel. Although the most accurate are the ones made of platinum, and therefore more commonly used, nickel or copper made RTDs are cheaper, while not as stable as the former. They are, however, the most expensive temperature sensors [55].

RTDs temperature range goes from -200 °C to 600 °C.

2.6.2 Thermocouples

Thermocouples are simple temperature sensors that are small sized, cheap, and are easy to use.

These sensors use two dissimilar metal wires that are welded together at one end, which is called a junction. Temperature changes produce a voltage difference between the two wires, causing an effect that is called the Seebeck Effect. This voltage can be measured and used to get the temperature values.

Thermocouple sensors are the ones that have the widest temperature range, which can go from -200 °C to around 1750 °C [55]. They can make use of vast materials for different temperature ranges accordingly.

However, thermocouples' output is small, and thus requires amplification. They are also not linear. Also, due to the thermocouple wires being different metals than the copper used for circuitry, another Seebeck Effect occurs on the junctions where these metals meet. Therefore, compensation is required for this effect in these junctions, that are called cold junctions.

2.6.3 Thermistors

Thermistors are temperature sensors that, similarly to RTDs, experience resistance changes with temperature variations [54]. They are usually made of ceramic semiconductor materials which use metal oxides, giving them fast responses to temperature changes.

There are two types of thermistors: Negative Temperature Coefficient (NTC), and Positive Temperature Coefficient (PTC). PTC thermistors' resistance value raises with temperature increasing, whereas NTC thermistors work the opposite way, which means that its resistance decreases with temperature increases.

These sensors are non-linear, although very precise. Their temperature range goes from -50 °C to 250 °C [55].

2.6.4 Semiconductors

Semiconductor temperature sensors are on integrated circuits and mainly based on temperature measurements using a transistor. By using its current and temperature characteristics, it is possible to convert the base-emitter voltage into temperature values.

These sensors have linear outputs, and they are cheap and small sized, but they have the narrowest temperature range, which goes from -70 °C to 150 °C [55].

The output values can either be analog or digital.

2.7 Humidity Sensors

Humidity is the amount of water vapor present in the atmosphere. It can be measured in either relative or absolute humidity [56].

Absolute humidity is the water content in the air. It is independent on the temperature and is given in grams of water vapor per cubic meter of air (g/m^3).

Relative humidity also measures the amount of water vapor, however, it is relative to the actual temperature, which means it is the ratio of water vapor in the air compared to the saturated level on the same environment conditions. It is measured in a percentage (RH or %RH). Relative humidity is the most often used unit, and sensors that measure it are often more accurate and cheaper.

In forest scenarios, high temperatures and low humidity levels can mean a high risk for fire outbursts and fast propagation, and therefore both of these variables should be related and taken into account. As so, only relative humidity sensors will be taken into account and studied.

2.7.1 Capacitive Sensors

Capacitive humidity sensors use a condenser that is humidity-dependent to measure water vapor levels, and producing almost linear outputs. They have a thin film composed of a polymer or metal oxide, which is put between two conductive electrodes [57]. The dielectric (thin film) constant varies with the relative humidity in the environment. In order to keep it from being contaminated, they are coated in either glass or ceramic. These sensors are accurate, and are functional at high temperatures.

2.7.2 Resistive Sensors

Resistive humidity sensors are composed of two metal electrodes that absorb water vapor, which results in electrical conductivity variations [57]. To avoid condensation and contamination, the sensors are coated. The impedance changes are related to humidity in an inverse exponential, thus needing linearization by either analog or digital methods. These sensors are temperature dependent, and are also prone to errors when exposed to condensation, thus being sensitive to contaminants. However, they are cheap, and small.

2.8 Enclosure

Outdoor applications should be designed taking into consideration the environment where they will be installed, as there are several factors that should be taken into account, such as weather hazards. Therefore, when developing outdoor electronics applications, an enclosure should be used that provides sufficient protection against different sorts of environmental hazards that can potentially damage the circuitry.

There are international standards that define the levels of sealing effectiveness of enclosures against a vast number of intruding elements, such as moisture, dust, or fire.

2.8.1 Ingress Protection

For moisture and foreign matter intrusion, it was developed the Ingress Protection (IP) rating (figure 2.24), which is an international standard that defines the levels of sealing for liquids, and solid objects. This protection is specified in IEC 60529: "Degrees of Protection Provided by Enclosures" [58], which was made a standard by the European Union.

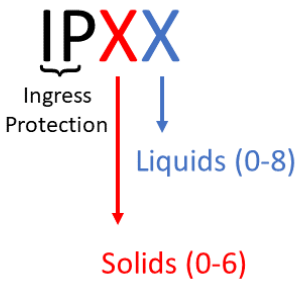


Figure 2.24: Ingress Protection rating standard nomenclature.

To define the rating, the nomenclature is composed of the Ingress Protection initials, followed by two numbers. The first number refers to the protection against solid objects, while the second against liquids [59]. Theoretically, the higher the rating, the better the protection, but also the more expensive it is. Therefore, the casing should be chosen carefully, taking into account where it will be placed, and what kind of protection will be ideal.

Table 2.4 refers to the IP rating against solids objects.

Table 2.4: IP rating against solid objects.

Rating	Type of protection
0	No specific protection.
1	Protection against foreign objects greater than 50 mm, such as a person's hand.
2	Protection against foreign objects greater than 12 mm, such as a person's fingers.
3	Protection against foreign objects greater than 2.5 mm, such as tools.
4	Protection against foreign objects greater than 1 mm, such as wires.

Continues on the following page

Table 2.4 – Continuation of the previous page

Rating	Type of protection
5	Protection against dust limited ingress.
6	Protection from total dust ingress.

Table 2.5 refers to the IP rating against liquids.

Table 2.5: IP rating against liquids.

Rating	Type of protection
0	No specific protection.
1	Protection against vertically falling liquid drops.
2	Protection against falling liquid drops when the enclosure is tilted at any angle up to 15° from its normal position.
3	Protection against direct sprays of liquid at any angle up to 60° from the vertical.
4	Protection against liquids sprayed from all directions.
5	Protection against ingress from low pressure jets of liquid, from any direction.
6	Protection against ingress from high pressure jets of liquid or heavy seas.
7	Protection against the effect of full temporary immersion in liquids. The test conditions for this rating are immersion in 1 meter of water for 30 minutes without any harm done to the contents of the enclosure.
8	Protection against long periods of immersion up to a certain pressure value that is specified by the manufacturer.
9k	Steam directed at a hight pressure against the enclosure from any direction.

2.8.2 UL 94 Flammability Standard

Since the case scenario that is being studied for this dissertation is the occurrence of wildfires in forest areas, and the devices will be placed in those locations where there is a possibility that

fires will reach them, it is important to study whether there are any materials that could possibly prevent or delay the fire's effect on the devices.

Although is might be impossible to stop all the effects that occur from fire presence, such as the high temperatures, which can permanently damage the circuitry and sensors, it is still relevant to prevent as much damage as possible.

The UL94 Standard for Safety of Flammability of Plastic Materials for Parts in Devices and Appliances is a flammability standard that was developed for polymeric materials. It defines a material's response to being in flames, which means its capability of either extinguishing or spreading flames.

There are a total of 12 classifications, 6 of which are specified for enclosures, and are represented in table 2.6.

Table 2.6: UL94 Flammability Standard

Rating	Type of protection
HB	slow burning on a horizontal specimen; burning rate < 76 mm/min for thickness < 3 mm or burning stops before 100 mm.
V-2	burning stops within 30 seconds on a vertical specimen; drips of flaming particles may occur.
V-1	burning stops within 30 seconds on a vertical specimen; drips of particles allowed as long as they are not inflamed.
V-0	burning stops within 10 seconds on a vertical specimen; drips of particles allowed as long as they are not inflamed.
5VB	burning stops within 60 seconds on a vertical specimen; no drips allowed; plaque specimens may develop a hole.
5VA	burning stops within 60 seconds on a vertical specimen; no drips allowed; plaque specimens may not develop a hole.

2.9 Conclusion

This chapter gave an overview of all the relevant topics that needed to be taken into account about WSNs and LPWANs, with particular emphasis on the LoRa technology and LoRaWAN protocol, as

this will be the LPWA technology used for this project.

It also showed all the different types of sensors that currently exist for each sensor that was chosen to be used, along with the main advantages and disadvantages of each.

Lastly, it was mentioned the two different standards for enclosures. Considering this project is to be used outdoors, it was important to mention the different ratings that exist and can be considered for choosing the best enclosure according to the requirements and constraints for this project.

Chapter 3

System Specification

After having a theoretical insight on the different technologies and aspects that this dissertation will focus on and having an overview of the different types of sensors that will be used, it is possible to define the components that will be part of the system.

First and foremost, it is important to outline that the application scenario that this dissertation covers is wildfire detection. However, on the long run, it is intended to have a functional generic node architecture that allows the use of different sensors without significant changes in its structure and coding.

This chapter presents the design and specification of the system and its architecture. All the requirements and constraints for the

A general overview of the whole node is given, taking into consideration the technological study made in section 2.

3.1 System Requirements

In order to properly design and conceive the system, it is necessary to define all its requirements and constraints beforehand, as it is crucial to fulfill them in the decision making process.

- LoRa should be used for wireless communications.

  It is intended to have a long range of network coverage in order to be able to place the nodes outdoors in remote areas.

- The nodes should be developed for low power consumption.

Since the system is to be battery powered, all the chosen hardware and developed software should take into account power efficiency methods, in order to provide the maximum lifetime.

- The system should measure a set of physical variables from the environment.

  The chosen physical variables to be monitored are: temperature, humidity, carbon monoxide, carbon dioxide, nitrogen dioxide, and light.

- Hardware enclosure should allow its placement outdoors.

  As the system is intended for outdoor applications, it should be enclosed in a way that it will not suffer from weather hazards. Its encapsulation should provide some sort of water and dust resistance.

- All the components should be chosen taking into account the overall budget of the system.

  Since it is intended to create a solution to integrate a Low Power Wide Area Network, then budget restrictions apply. In order to be able to produce a vast number of devices, their individual cost should be the lowest possible.

3.2 System Architecture

As mentioned in the preceding sections, it is intended to develop a node architecture that will allow collecting environmental data and sending to the upper levels of a Low Power Wide Area Network. Taking into account the all the requirements and the study that was previously made in chapter 2, it is possible to define the system components. Firstly, the system is divided into two major layers: the hardware layer, and the software layer. The full system stack, containing all the different layer divisions and elements, is depicted in figure 3.1.

The different elements that compose the hardware layer of the system are the microcontroller, the different sensors, and the radio module. The chosen sensors are a temperature sensor, a humidity sensor, a light sensor, and a gas sensor.

Above this, there is the software layer. It is the microcontroller that will interface the hardware and the software layers, and thus will contain all the software elements. The software layer can be split into two different sub layers: the software low layer, and the application layer.

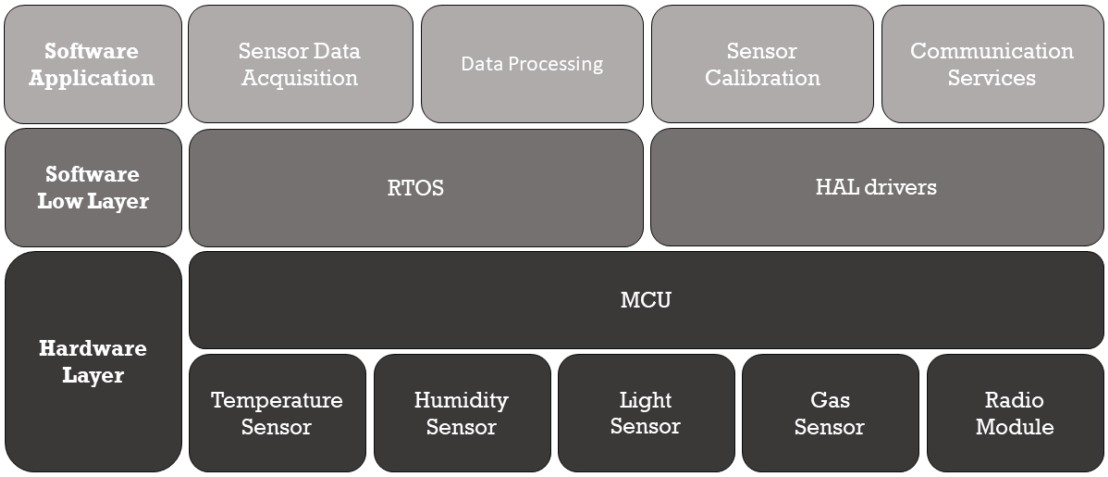


Figure 3.1: System stack, containing both the hardware and software layers their respective elements.

The software low layer contains the software fundamentals for the MCU's peripherals and functioning, the Hardware Abstraction Layer (HAL) drivers, and the Real Time Operating System (RTOS). Over this sub layer, the software application is responsible for the functioning of the edge devices. Thus, it contains the sensor data acquisition and processing, as environment information needs to be collected and have some sort of processing in order to know if the collected values are within expected or acceptable values. This sub layer also contains the sensor calibration methods for when this procedure is possible, and the communication services that are relative to the radio module, as they are fundamental for the nodes' role in the LPWAN.

3.3 Hardware Specification

After specifying the system requirements that can be seen in section 3.1, and having defined system stack and its different hardware layer elements, it was then possible to make a selection of what components to use on the node for the case scenario that this *Master's thesis* focuses on, in order to provide reliable data to the upper levels of the LPWAN. This section focuses on the components chosen for this dissertation.

3.3.1 Microcontroller

For the purpose of this dissertation, it was intended to be used an ARM MCU. Since one of the requirements is energy efficiency, from the Cortex ARM MCUs family, the type that better suits

these needs are the M0+ microcontrollers, which are built with the intents of being small and having low energy consumption.

The chosen MCU was the STM32L081CZ. It belongs to ST's 32-bit *Ultra Low-Power* group of Cortex-M0+ based MCUs [60], the STM32L0 family, which are aimed at battery operated devices and applications with budget constraints.

It contains a set of low power modes that allow optimizing the power consumption, as well as wake up sources, and some low power peripherals, such as low power timer (LPTIM), and low power UART (LPUART).

Some of the STM32L081CZ's main characteristics are:

- 128-Kbytes Flash
- 6 Kbytes of data EEPROM
- 20 Kbytes of RAM
- Backup register with 20 bytes
- 1.65 V to 3.6 V power supply
- Temperature range: -40 to 125 °C
- USART, I2C, and SPI interfaces
- 12-bit ADC
- 7-channel DMA controller
- 11 timers

One of the main advantages of these processors is the compatibility with the rest of the Cortex-M family, as it is relatively easy to port applications from one MCU to another.

3.3.2 Gas Sensor

Wildfires have a direct connection to air quality, as seen in the previous chapter. From the gases mentioned in section 2.5, where a study was done to find the most viable sensors for this case scenario, it was chosen to monitor carbon monoxide, carbon dioxide, and nitrogen oxide.

For these components, different types of sensors exist, as mentioned in section 2.5.4. From the studied gas sensors, the ones that showed the most promise in gas detection were the NDIR sensors, due to their capability of distinguishing gases based on their absorption behaviour of infrared light, as well as their characteristic of being less prone to contamination than others.

3.3.2.1 Carbon Dioxide Sensor

For CO_2, it is mostly used infrared sensors for its detection, as this gas has a very characteristic absorption band, hence making these sensors very effective for this purpose.

The sensor used for carbon dioxide was the Cozir sensor (figure 3.2).

Figure 3.2: Cozir Carbon Dioxide Sensor.

It is an NDIR sensor (as seen previously on section 2.5.4), that has a measurement range of 2000 ppm [61]. Some of its characteristics are:

- 3.3 V, $<$ 1.5 mA average power requirements
- Low noise measurement (<10 ppm)
- Peak current 33 mA
- 2 measurements per second
- Serial communication 9600/ 8/ 1/ n
- Analog voltage output proportional to CO_2 concentration
- SHT21 Temperature plus humidity sensors built-in (serial output only)

- Measurement Rate: 0.5 sec / measurement
- Accuracy: ±50 ppm ± 3% of reading

3.3.2.2 Carbon Monoxide and Nitrogen Dioxide Sensor

Although CO is also measurable with infrared sensors, this was not a viable option due to their high price, and this type of sensors require high consumption to detect carbon monoxide, which does not meet the requirements for this project. Electrochemical and resistive sensors are also able to detect fairly well this gas.

Resistive sensors are good at reducing or oxidizing volatile organic compounds or toxic gases, like nitrous oxides [52]. Electrochemical sensors are also a good option to detect nitrogen dioxide. Therefore, the sensor choice inferred mostly on these two types of sensor.

As electrochemical sensors are more expensive and are more prone to contamination from the environment, it was decided to use a resistive sensor to detect NO_2. These are cheaper, and although they can be influenced by temperature and they can have false positives, it is expected that calibration methods can be used to overcome the environment influence.

For nitrogen dioxide and carbon monoxide, it was chosen the MiCS-4514 sensor (figure 3.3). This os a MEMS metal oxide semiconductor sensor, that can detect both of these components.

Figure 3.3: MiCS-4514 Carbon Monoxide and Nitrogen Dioxide Sensor.

To detect the presence of the gases, the sensing resistance is measured, where the reducing (RED) sensor resistance **decreases** when CO is present, whereas the oxidizing (OX) sensor resistance **increases** when NO_2 is present.

Some of this sensor's main characteristics are:

- Two gas sensors in one small sized package
- Short pre-heating time

- 5 V power supply voltage
- RED sensor sensing resistance in air ranging between 100 and 1500 kΩ
- OX sensor sensing resistance in air ranging between 0.8 and 20 kΩ
- Detection range for carbon monoxide between 1 and 1000 ppm
- Detection range for nitrogen dioxide between 0.05 and 10 ppm
- Typical heating current of 32 mA and 26 mA for RED sensor and OX sensor respectively

3.3.3 Temperature and Humidity Sensor

The temperature and humidity sensor's choice was restricted due to the carbon dioxide sensor. As this component had already an integrated a temperature and humidity sensor by itself, it was important to make a similar choice, and thus it was decided to use the same component, in order to have similar readings.

The chosen sensor, SHT21 (figure 3.4), integrates both relative humidity and temperature measurements into one component. It contains a capacitive humidity sensor, and a band gap temperature sensor, which is a specific kind of semiconductor temperature sensor.

Figure 3.4: SHT21 Temperature and Humidity Sensor.

Some of this sensor's main characteristics are:

- Temperature and humidity sensors in a single component
- Small sized CMOS chip
- Digital I^2C interface
- Humidity sensor resolution: 12 bits

- Temperature sensor resolution: 14 bits
- Operating range: -40 to 125 °C
- Maximum measuring current: 330 μA
- Device address 1000 000x

3.3.4 Light Sensor

In order to help fire detection, a light sensor can be used. It can be extremely relevant especially when monitoring parameters at night, as fire will produce more light than usual during night time.

The light sensor was chosen taking into account previous sensors used in other projects. The model that was chosen is the MAX44009, which contains a photodiode that converts the light into a current. It has a very small consumption, and provides a wide range of light measurements. It is able to optimize the measurements by adapting to either dark or bright situations, in order to provide more viable readings. The sensor tries to mimic the human eye behaviour, and blocks IR and UV lights by using an optical filter.

The chip already contains an ADC, which helps turning the read signal into a digital bit stream htat is then processed and stored, so it can be read by an I^2C interface. This sensor also provides a programmable interrupt, which enables readings without constantly polling it.

Some of this sensor's main characteristics are:

- Small sized
- Measurement range: 0.045 to 188,000 lux
- Digital I^2C interface
- IR and UV blocking capability
- Operating range: -40 to 85 °C
- Maximum current: 1.6 μA
- Device address: 1001 010x or 1001 011x

3.3.5 LoRa Module

The radio frequency module that was chosen is Microchip's RN2483 (figure 3.5). This RF module is certified for operations in 433 and 868 MHz, and uses the LoRa technology. It contains the LoRaWAN protocol in order to be able to connect to a LoRa network, and enables class A devices. Its certification is issued for LoRaWAN 1.0.

This was one of the first LoRa modules on the market. There is an equivalent for the 915 MHz frequency, the RN2903

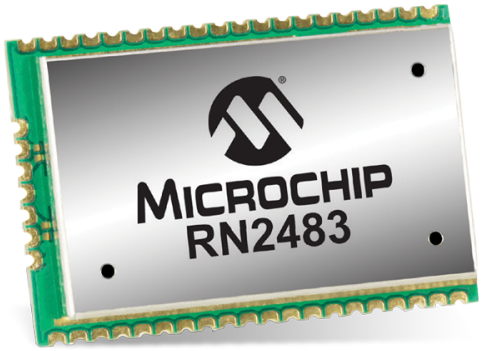


Figure 3.5: RN2483 LoRa module. (Source: [62])

Some of the RN2483's main characteristics are:

- On-Board LoRaWAN Protocol Stack
- UART interface
- ASCII Command Interface
- Device Firmware Upgrade (DFU) over UART
- Low-Power Consumption
- 14 GPIOs available for user control, some with Analog Input functions
- Operating temperature ranges: -40°C to +85°C
- Long Range Transceiver able to operate in the 433 MHz and 868 MHz frequency bands
- FSK, GFSK, and LoRa Technology Modulation

- Manufacturer's defined range: up to 15 km coverage in areas; up to 5 km coverage at urban areas
- Adjustable Transmission Power up to 10 dBm for 433 MHz and 14 dBm for 868 MHz

Inside this module, a set of components can be found from which it is relevant to specifically mention the transceiver and the microcontroller. The module's block diagram can be seen in figure 3.6.

The host MCU represents the microcontroller that is chosen for the node, which is chosen by the user. It will control the application, and communicate with the RN2483 through the UART interface. All configuration for the LoRa network can also be stored in an EEPROM available in the module, which allows developers to preset some parameters, such as the sessions keys mentioned in section 2.4. The mentioned transceiver (Lora-Compliant Radio), and the RN2483's microcontroller (MCU) can be seen in the diagram as well, and since this is the European version, the 433 MHz and 868 MHz frequency bands are supported. The module can use two antennas, one for each frequency band, although it can only operate in one at a time.

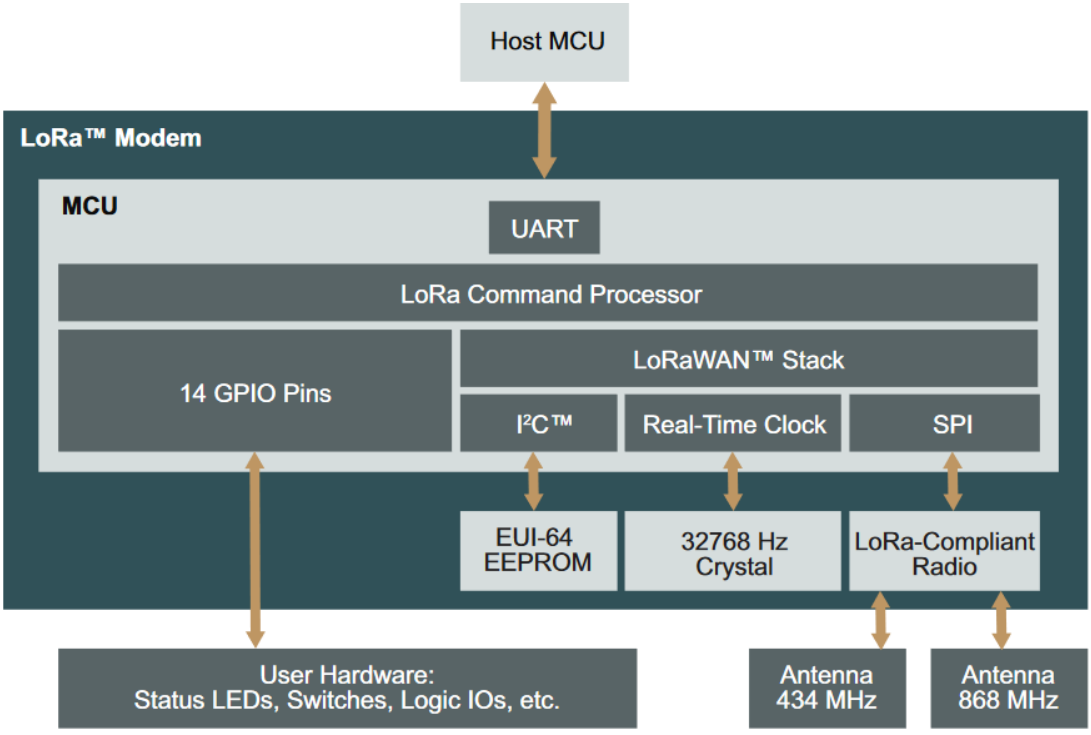


Figure 3.6: RN2483 LoRa module block diagram. (Source: [63])

This LoRa module contains a Semtech SX1276 transceiver, which uses the LoRa modulation technique, and also offers support for FSK transmission [38]. This transceiver can operate in frequencies between 137 MHz and 1020 MHz, thus the frequency choice should be region specific. It allows the choice of bandwidth from 7.8 kHz to 500 kHz, and the spreading factors from 6 to 12 for all frequency bands. The SX1376 can achieve a high sensitivity of -148 dBm.

Exclusively for LoRa communications, a First In First Out (FIFO) buffer with a size of 256 bytes is used. It is located in a RAM area, which is fully customizable by the user and provides access to all the received or transmitted data. All data from the last reception operation is available until it transitions to receive mode or when it enters SLEEP, which is when the FIFO data buffer is automatically cleared. By default, the SX1276 transceiver assigns half of the available memory for transmit data, and the other half for received data.

All access to this data is done by the SPI, and thus even all configuration registers are set using this interface. The transceiver timing is given by a crystal oscillator, which is required.

The RN2483 also has a PIC18LF microcontroller which allows to interface the SX1276 transceiver to a host MCU through the UART interface, and contains the LoRaWAN stack. By allowing a set of ASCII commands to optimize the command/response interface with a host MCU or system, this RF module makes applications' development easier and quicker. It also provides a set of 14 GPIO Pins, which can be configured by the user or, in some cases, be used as analog inputs.

The PIC microcontroller also offers the possibility of updating its firmware through the UART, or by using the In-Circuit Serial Programmed (ICSP) pins. At the time of the writing of this dissertation, the latest available firmware was version 1.0.4, which was released on the 11th of February of 2018, and implemented the LoRaWAN 1.0 protocol [64].

3.3.6 Battery

The chosen battery to power the nodes is the SAFT LS14500. It is a lithium AA sized battery which is provides 3.6 V [65]. One of its main characteristics is the fact that this battery keeps its voltage very steady for the most part of its life time. Its capacity is 2600 mAh.

Figure 3.7: Chosen battery. (Source: [65])

3.3.7 Enclosure

After selecting all the components and in oder to make it possible to do the hardware implementation, it was necessary to decide which kind of enclosure to use for the nodes. Considering that the nodes will be placed outdoors, they will be prone to weather and other sorts of environmental hazards, and after studying different standards for enclosures in section 2.8, it was possible to choose the enclosure which will best fit the dissertation's purpose.

Therefore, after analysing the standards, it was decided that the enclosure should be totally protected against dust (level 6 of the IP rating for solids), and at least protection against low pressure jets of liquids (level 5 of the IP rating for liquids), thus the enclosure having a minimum rating of IP65 would be ideal.

The enclosure should also provide some sort of flame protection. Even though it will not completely stop the harmful effects of fire in the node's electronics, as the components might malfunction after being exposed to the high temperatures that occur with wildfires, it should at least a V-0 rating on the UL94 standard.

After taking all the previous things into consideration, and further checking suppliers, the best fitting choice was the plastic 1554E2GYSL case with the dimensions 89.99mm x 89.99mm (length and width, respectively), manufactured by Hammond [66], which provides the rating IP66, and UL94 V-0. Due to coherence in the measurements when compared to the casings used in previous projects, these dimensions were the ideal, since they should be more than enough to fit all components.

However, the pricing for this case was relatively high for the type of application that was intended, since a vast number of nodes is to be produced. Therefore, an option had to be made

Figure 3.8: 1554EGY, the chosen enclosure. (Source: [67])

for disregarding one of its characteristics, in order to achieve a lower price. As so, the final enclosure choice was also a plastic Hammond box, the 1554EGY (shown in figure 3.8), with the same dimensions, although with a lower flammability rating [67]. Since, as previously mentioned, it may not be possible to prevent fire damage due to the high temperatures, it was decided that this parameter would not be as relevant for the node as the protection against dust or liquids.

Since the enclosure is completely sealed off, it could bring some issues not only for an accurate gas detection, but also for the temperature and humidity measurements. These sensors are meant to measure the environment variables, and being completely sealed off from it might cause wrong values. Hence, it was relevant to find some sort of system that would allow airflow within the case, as to provide a temperature inside the casing as similar as the one outside, and also the same air composition and pressure.

This can be done by installing vents in the case, which can also come with IP ratings [68]. A protective vent installed on the case may allow necessary airflow, as well as prevent heat dissipation, while still maintaining the sealing requirements. Therefore, a small protective vent was chosen with a 17.1 mm diameter (figure 3.9), which should be able to fit and be installed on the chosen enclosure [69]. This vent not only allows the airflow, but also has an IP68 rating.

Figure 3.9: Protective vent to be installed on the enclosure. (Source: [67])

3.4 Conclusion

This chapter started by taking into account the main goals and the requirements, and it was further possible to analyse and build a system stack, which allowed to better choose the components. After choosing the microcontroller and the sensors, a brief overview of each was given, having in mind the application scenario. Since this project aims to build a node architecture for outside placement, it was also chosen an enclosure and some ways to prevent the environmental hazards to the electronics that it will contain.

After choosing all components, it was then possible to proceed to making the circuits schematics and starting the hardware development, as well as the software development that would support it.

Chapter 4

Implementation

After defining all the system specifications and components that were used, it was possible to proceed to the implementation in order to develop the nodes and fulfil the dissertation's objectives.

Therefore, this chapter provides the overview of how the system came to be. It is divided into two main sections that are relative to the hardware and software components of the system.

The hardware was developed taking into consideration a general node architecture, which would make use of generic available peripherals in order to contain any type of sensors. In order to be able to test this concept, it was decided to develop two variants of the node architecture. They contained different sets of sensors, thus allowing to test and see if the concept was properly developed.

The software was developed accordingly with the hardware layer, as it was necessary to develop all the code for all the peripheral interfaces. It was also necessary to take into account that the generic architecture could be used for different variants of the nodes, and thus it was developed accordingly for the two node variants, in order to further prove this concept.

4.1 Hardware Implementation

This section contains the development process for the hardware. Firstly, it was chosen a general node architecture, by using the microcontroller's graphical tool that allows mapping the peripherals and pins. Afterwards, it is shown how both variants were built and the necessary circuitry that was added.

4.1.1 Node Architecture

The general node architecture was developed regardless of the sensors to be placed on the board. There were two variants of nodes that were developed for this *Master's thesis*, which are explained in the following sections.

It was decided to use as a general set of components, the MCU and the LoRa module. However, the enclosure that had been chosen in section 2.8 was not used. As this was a first prototype, it was opted instead to use an already available enclosure, which had already been used previously in other projects, in order to reduce the budget. The box used has an IP54 rating, and its dimensions are 100 mm x 100 mm.

Since the considered sensors will have different peripheral interfaces, and they are not to be used unless when reading environment data, it was decided to use a GPIO pin to switch them on or off, accordingly. This decision was also made for the LoRa module, ir order to allow testing different approaches for node development, taking into account the two different authentication methods. To define where each peripheral would be assigned, as well as the needed GPIO pins for enabling or disabling sensors and the LoRa module, the *STM32CubeMX* tool was used. This software tool allows to map and set all the peripherals and GPIOs within a given STM32 MCU, and thus helps to prevent peripheral overlapping peripherals.

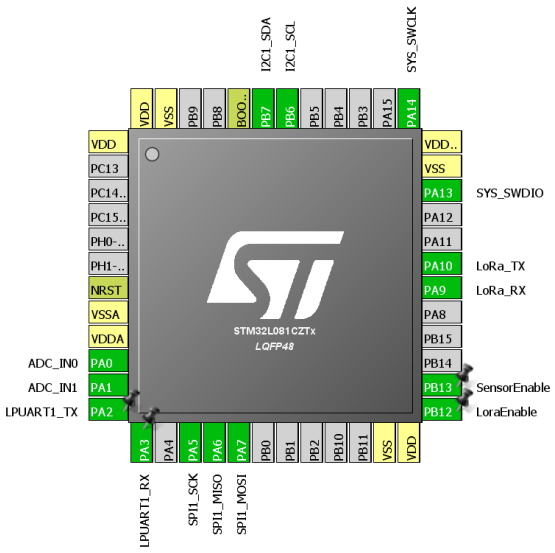


Figure 4.1: Chosen pinout for peripherals (using *STM32CubeMX* tool).

The chosen pin mapping, which can be seen in figure 4.1, is representative of how the node architecture was chosen. For different variants of nodes, peripherals can be activated through

software implementation, depending on which sensors there are. This will be further explained in the *Software Implementation* section.

Similarly, it was used the MCU's *USART2* for the RN2483 module, and the *I2C1* for the I^2C sensors, as well as one *USART (LPUSART)*, the SPI interface, and two ADC channels.

Since the considered sensors will have different peripheral interfaces, and they are not to be used unless when reading environment data, it was decided to use a GPIO pin to switch them on or off, accordingly.

The sensors and the LoRa module are both activated by a specific GPIO, which was set as an output that can be changed to turn them on or off. For the LoRa module, it was named *LoraEnable*, and chosen the pin *PB12*. For the sensors, the GPIO output was named *SensorEnable*, and assigned to pin *PB13*.

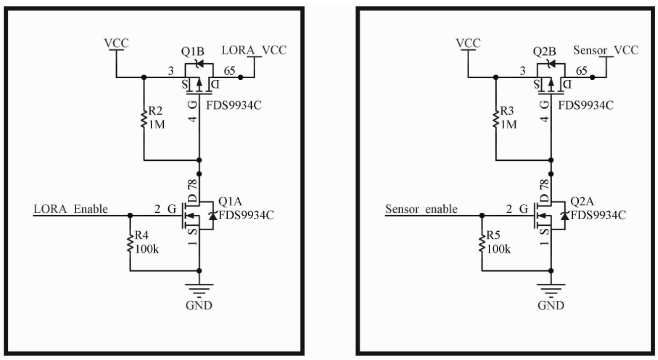


Figure 4.2: *LoraEnable* and *SensorEnable* circuitry.

These pins are connected to two MOSFETs, which will give the voltage output to the peripherals, as seen in figure 4.2. When enabled, they will be directly powered by the 3.3 V that is supplied to the board by the low-dropout regulator (LDO).

When the enable pin is at a low state, the first MOSFET (enhancement mode N-channel) will have a Gate-Source voltage (V_{GS}) of 0 V, making the second MOSFET's gate have the value of VCC. Thus, on the enhancement mode P-channel MOSFET, its V_{GS} will also be of 0 V, and the sensors will not be powered.

When the opposite occurs, which means that the enable pin is at a high state, the N-channel V_{GS} will have a positive value that exceeds the Gate Threshold Voltage, thus making the gate of the P-channel MOSFET have a low value. This will make the Gate-Source voltage of the P-channel MOSFET be negative, and then the sensors will be provided the same voltage as the VCC.

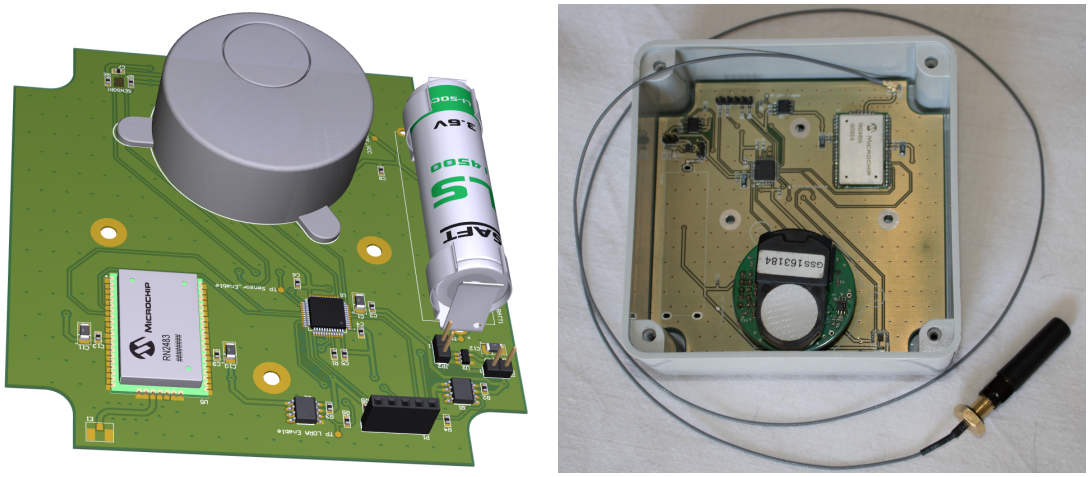
Both resistors that are in this circuit are to make sure the pull-down / pull-up are done correctly when the output is off. By using this circuitry, it was possible to reduce the power consumption, as the sensors are only turned on when readings are needed.

As mentioned in the previous chapter, LoRaWAN's OTAA authentication method requires the LoRa radio to be on at all times for it not to lose the session. Even though this is a constraint, it was still added the enable pin, as it can be used for ABP, or to reset the module, and thus the same circuitry was used for this purpose.

4.1.2 Variant 1

The first variant of the nodes contained just two sensors and thus was rather simple to design. It used the Cozir CO_2 sensor, which already contains the SHT21 temperature and humidity sensor, as seen in section 3.3. This sensor was assigned to the *LPUART*, as its baudrate was enough for this peripheral. As there was no need to add a SHT21 sensor, the I^2C line contained only the light sensor. As mentioned in the previously, the *USART2* was also used for the LoRa module. All the other peripherals were deactivated.

This node's board design can be seen in figure 4.3. The schematics and Printed Circuit Board design can be found in Appendix A.



(a) 3D board design. **(b)** Board and its box.

Figure 4.3: Final result of the node variant 1

4.1.3 Variant 2

This node contains a light sensor, a SHT21 temperature and humidity sensor, and a gas sensor. Both the light sensor and the temperature and humidity sensor on the same I^2C line. This is possible as they both have different addresses, thus not requiring extra peripherals. *USART2* is also reserved for the LoRa module. For the gas sensor, since it integrates basically two sensors in one component, two distinct channels from the same ADC peripheral were used.

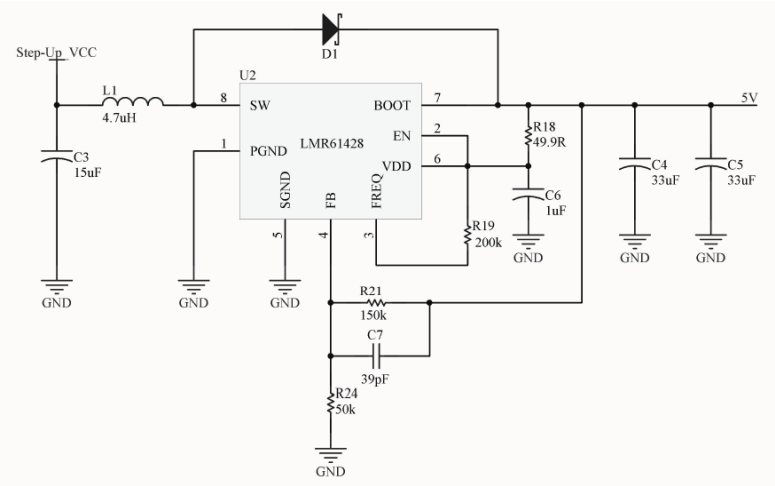


Figure 4.4: LMR61428 Step-Up voltage regulator and its circuitry.

Since the carbon monoxide and nitrogen dioxide sensor operates in the 5 V supply range, and the chosen microcontroller operates in the 3.3 V, a converter was needed to have a higher voltage. Therefore, a Step-Up converter was used to provide the right voltage to the sensor and ensure it works. The Setp-Up voltage regulator used was the LMR61428, which can be seen in figure 4.4. The setup used provided an output of 5 V and 0.2 A.

The converter is switched on whenever a reading needs to be made. This is done with the same type of circuitry as the other sensors, shown in figure 4.5. Although it was used the same *SensorEnable* GPIO pin to turn the MiCS-4514 sensor on or off, it was used another set of MOSFETs, as its input voltage was different than the remaining sensors on the board.

In this case, the input voltage for the Step-Up converter is provided by the battery in order to avoid unnecessary voltage level adjustments, as the voltage is lowered by the LDO, as previously mentioned.

Figure 4.6 shows the circuit schematic for the MiCS-4514 sensor. It was divided into two parts: the heating circuitry, and the sensing circuitry. The sensor's heaters need specific resistance

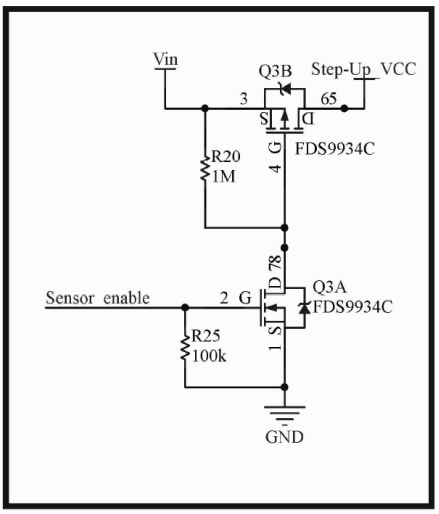


Figure 4.5: *SensorEnable* circuitry for the Step-Up converter.

values, which are 82 Ω for the RED sensor, and 133 Ω for the OX sensor.

The sensing circuitry includes a *Load Resistance*, which is specified by the manufacturer that it must a value of at least 820 Ω in order not to damage the sensing layer of the sensor. This resistance is what is used to measure the sensor resistance changes, by measuring its voltage drop. For this, the microcontroller's ADC was used, one channel for each of the sensors. This required having special care when designing the Printed Circuit Board (PCB), and thus a separate zone without a ground plane was made to keep the MiCS-4514 without interference.

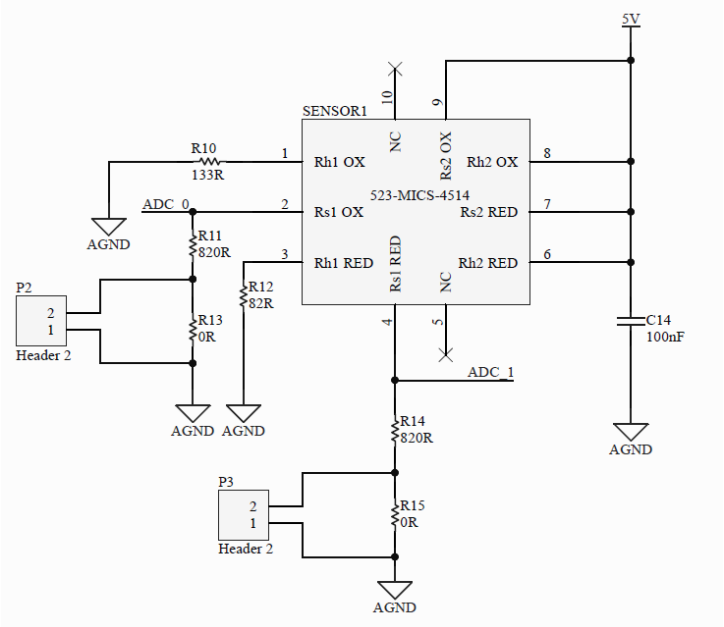
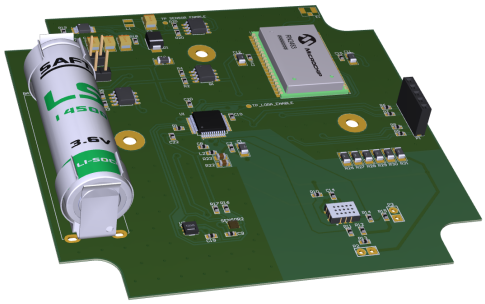


Figure 4.6: MiCS-4514 sensor circuitry.

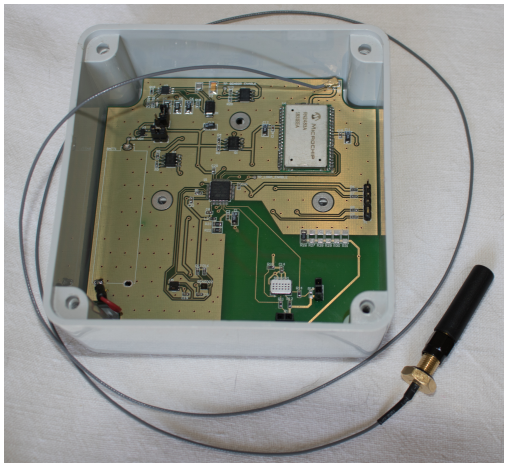
For this sensor's operation, it was necessary to define the *Load Resistance* in order to be able to get accurate readings from it. Since the manufacturer's documentation is not very specific in this matter, further experimental adjustment was required to choose the resistance value.

Therefore, when designing the circuitry for this node variant, it was placed a header for each *Load Resistance*, in order to experimentally find the best value.

The node's final design can be seen in figure 4.7. The schematics and Printed Circuit Board design can be found in Appendix B.



(a) 3D board design.



(b) Board and its box.

Figure 4.7: Final result of the node variant 2.

4.2 Software Implementation

This section contains the steps taken for the software development. It describes the development environment used, how the boards were programmed, and the different software layers.

4.2.1 Development Environment

The nodes' software was developed in the Integrated Development Environment (IDE) *Keil µVision*, using the *C* programming language. This IDE allows to develop, manage, build and debug embedded software. It supports a wide set of devices.

To manage and choose the pin mapping and assignment of all the peripherals, it was used the *STM32CubeMX* tool [70], which allow a graphical configuration of the STM32 MCUs. It allows setting each peripheral individually, map their pins to the available options, and even manage the

embedded operative system. The provided code initialization was given in the C language, and the project was specifically created for the *Keil µVision* IDE.

As previously seen, code can be developed for a STM32 MCU and easily ported to a different one. As so, a NUCLEO-L073RZ board [71] was used to develop all the code before the node boards were assembled and able to be programmed. This board's MCU (STM32L073RZ) is very similar to the one chosen to the project, thus it was adequate for code development. The code was then ported to the target MCU, and the board was programmed through the use of the Nucleo board.

The Nucleo board contains an on-board programmer and debugger, the ST-LINK/V2. This allows not only to program the Nucleo board itself, but also to program external MCUs [72]. This programmer contains JTAG and Serial Wire Debug (SWD) pins.

To program the nodes, it was used the SWD interface, along with some external power pins that are necessary for a successfully program the target MCU. Figure 4.8 shows the Nucleo board, with the ST-LINK/V2 part of the board shown in blue. The SWD pins are shown in orange, and in red are the two jumpers, which dictate the target chip to be programmed. On the one hand, with the jumpers on the board's microcontroller is programmed, on the other hand, they have to be removed so an external MCU can be programmed.

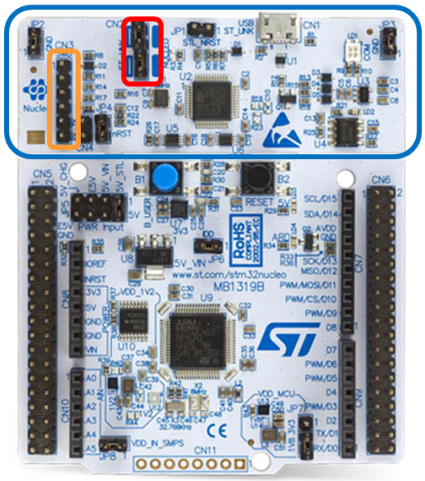


Figure 4.8: STM32 Nucleo-64 board and its on-board debugger.

After connecting the boards, it was then used the STM ST-LINK Utility [73], which is a software interface to program these MCUs. This tool allows to write, read or read a device's memory, as well as using Printf through the SWO pin in the SWD interface. However, the chosen STM32 MCU does not provide the SWO pin.

4.2.2 Software Layers

As seen in the previous chapter on figure 3.1, the software is divided into two layers: the Software Low Layer, and the Software Application. They are explained below.

4.2.2.1 Software Low Layer

The Software Low Layer is composed of the Hardware Abstraction Layer (HAL) drivers and the embedded Real Time Operating System (RTOS).

The HAL drivers provide a set of Application Programming Interfaces (APIs) which allow the interaction of the developed application to make use of the peripherals in the MCU. They allow peripheral configuration and initialization, interrupt handling, or data transfers, while being portable to other MCUs as well.

The *STM32CubeMX* tool supports the use of FreeRTOS, allowing its configuration. FreeRTOS is an open source real time operative system. It is especially designed having in mind microcontrollers, although it is not only limited to their use. Since it is officially supported by ST, it was integrated in *STM32CubeMX* and it can be configured from there. Over the operating systems, ST developed an API called CMSIS-RTOS [74]. It provides an interface that can be ported through several RTOS and contains a set of libraries, templates and other features. CMSIS-RTOS is basically a wrapper for any RTOS, thus allowing to swap the base RTOS used without having to do major changes on the project. CMSIS-RTOS comes by default on the *STM32CubeMX*, and it was used over FreeRTOS.

4.2.2.2 Software Application

Over the software layers provided by the MCU manufacturer, it was developed the intended software application for the nodes. The core concept was to develop all the code for all the peripheral interfaces.

To understand how the node's software was developed it is first necessary to mention again what the purpose was. The nodes' main goal was mainly to monitor environment variables and further sending that information to the upper levels of the network.

Overall System Workflow

In order to make it possible to properly communicate with all the sensors and the LoRa module, it was necessary to develop a set of APIs that would provide an abstraction level to the HAL provided by the microcontroller's manufacturer. Hence, each peripheral had its own set of APIs, which enabled an easier development for the sensors. The peripherals were, as previously mentioned, the two USARTs (one for the LoRa module, and another for a sensor), the I^2C, the SPI, and the two ADC channels. These set of APIs can be easily changed and fit to handle more peripherals if needed. Furthermore, they made it possible to swap the sensors and their respective code without changing how the peripherals work.

Following the main workflow (as seen in figure 4.9), when the system starts everything is initialized, starting with the HAL and the system clock, followed by all the necessary peripherals. The used peripherals are given by conditional compiler file containing a set of defines, which specify which sensors and peripherals are used.

When the system is firstly initialized, all the necessary peripherals to interface the sensors are initialized according to the pre-specified values for each sensor in the *DEFINES* header file. After all peripherals are initialized, a checkup is done to all the sensors, according to the given code for the sensors. For these application cases, it means sending a "dummy command" to the Cozir sensor that returns a simple answer, or checking through the I^2C line the device addresses of the SHT21 and the MAX44009 sensors to see if they are connected. The LoRa module is also checked following the same procedure as the Cozir sensor.

Afterwards, the RN2483 LoRa module is authenticated within the network. This requires that all the LoRaWAN keys must be stored in the device beforehand. This is possible in the RN2483 module because it contains an EEPROM that allows storing parameters or even the session keys (as mentioned in section 3.3). If the module is successfully authenticated within the network, it will follow the flow of its course. For the purpose of this project, the authentication method used was the ABP.

After all peripherals and components are initialized and checked, the operating system is then configured, thus allocating all the necessary resources for its tasks, as well as the task synchronization methods. The scheduler will then take manage the system, and the FreeRTOS will be started.

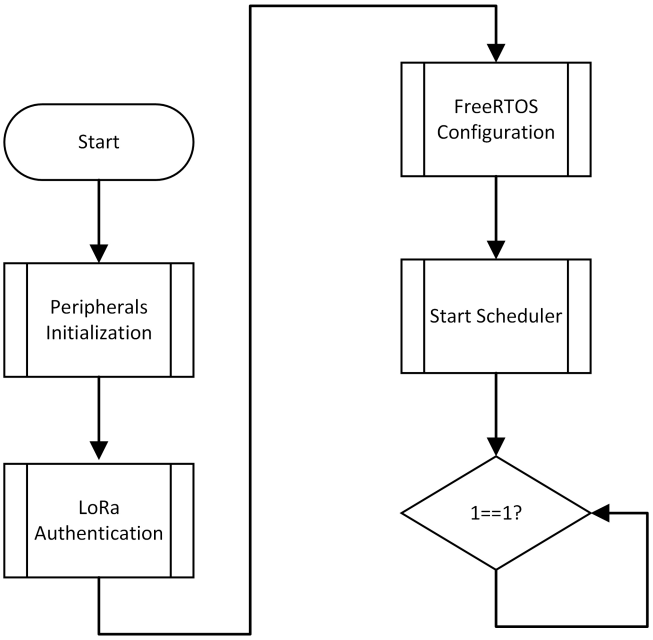


Figure 4.9: System startup flowchart.

The system contains three main tasks: the Data Acquisition task (*vDataAcquisition*), the Data Sender task (*vDataSend*), and the Data Process task (*vDataProcess*).

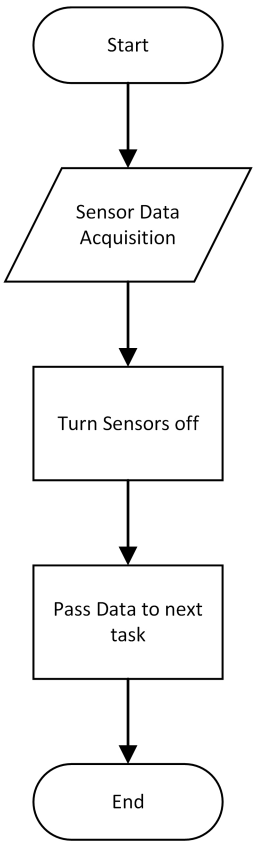


Figure 4.10: Data Acquisition task flowchart.

The data acquisition (figure 4.10) task is mainly responsible for getting the data out of the sensors. It is this task that retrieves the data for the sensors and saves the collected values. After the data is read, the sensors are turned off through the *SensorEnable* pin. The data is then forwarded to the next task, which is responsible for processing it.

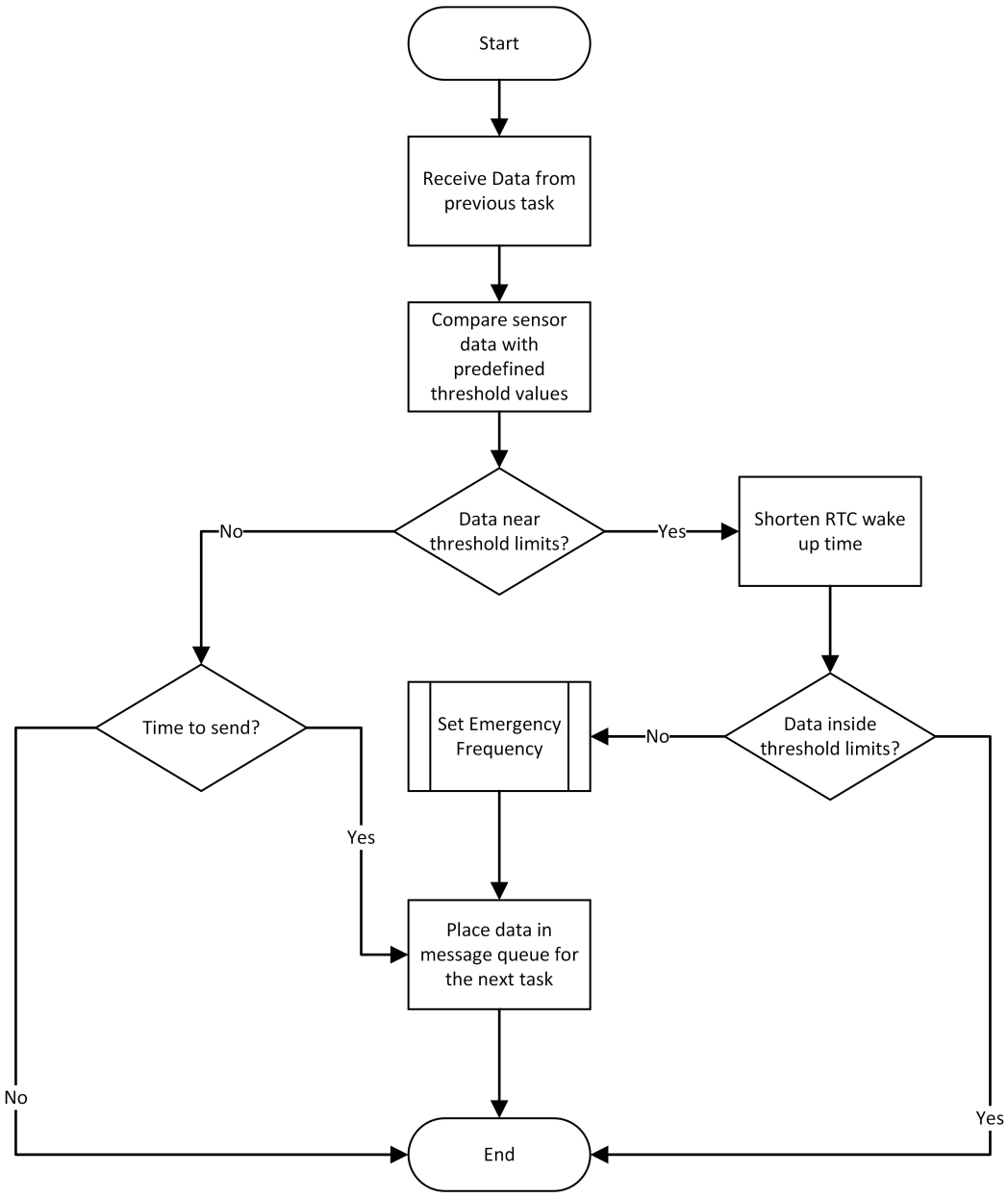


Figure 4.11: Data Process task flowchart.

When developing a CPS, some parts of data processing occur on the lower levels, rather than just acquiring data and sending it over and over. As this would cause a high consumption due to the LoRa module transmission, especially when at lower bit rates, data is only sent, for the most part, when the values are outside of some predefined limits. This allows that the upper layer can

make predictions based either on the weather forecast, or the season, and send that information to the node, so the acceptable values for each sensor threshold can be adjusted.

The data process task receives the information previously collected on the previous task, and will compare the values with those predefined limits. Data will be sent if it is a scheduled transmission, or if the values are outside of the threshold limits. In the latter, the node is assigned a specific frequency, called the *emergency frequency*. Since the gateway does not have access to the data, this is a way for it to know that an important message is being sent. Since the nodes work with 8 channels, to force the *emergency frequency*, all channels except one are disabled, and that channel is forced at the *emergency frequency*.

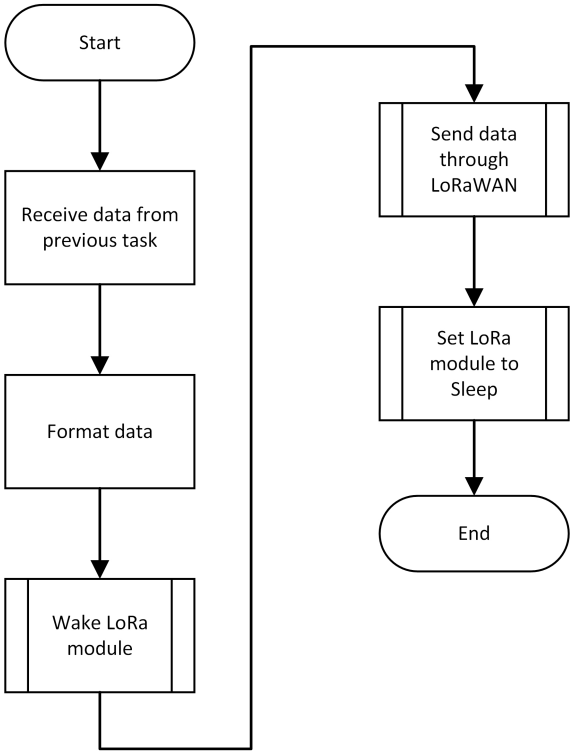


Figure 4.12: Data Sender task flowchart.

If data is supposed to be sent, it will be forwarded to the next task, which is the sender task. This task is responsible for formating the data, and send it through the LoRa module. If the authentication method is ABP, then the LoRa module will be turned off. However, if the authentication method is OTAA, then the LoRa module will have to be permanently on, and thus will only be placed in *sleep mode* after sending the message.

Low Power Mode

In order to optimize and reduce the power consumption, the nodes are set to a low power mode when they are not acquiring or sending data. This is possible due to the manufacturer [75] The chosen MCU has seven different low power modes.

Ideally, the Standby mode should have been used, as it has the lowest power consumption. Since it has a variant that can be awaken using the RTC, this would be the ideal implementation. In this mode, only the LSE or LSI oscillators are running, and the CPU, Flash memory, and RAM memory are deactivated. Only the RTC back up registers are available, which is a total of 20 registers. However, due to the hardware implementation of the nodes, and having in mind that the LoRa module needs to stay on the whole time due to the OTAA session, this was not fit for this application, as the hardware peripherals are switched off in this mode.

Consequently, the best mode to be used was the Stop mode, which keeps the RAM memory and the RTC backup registers, hence stopping the CPU. Similarly to the Standby mode, this one is also able to wake up from the RTC, although it can also be awaken from other peripherals such as the USART, I^2C, Low Power UART, or the Low Power timer.

It was developed a module to contain the internal RTC initialization, its configuration, the activation and configuration of the Wake Up and deactivation of the Wake Up. The RTC synchronous prescaler was set to 1 Hz to provide a 1 second minimum timebase/wake up time, and a maximum of 18 hours.

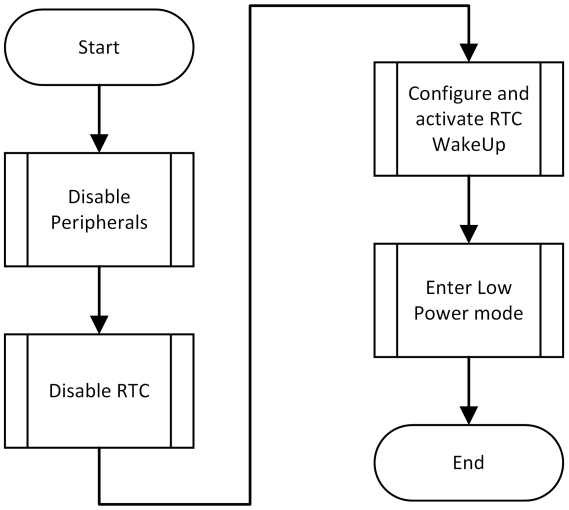


Figure 4.13: Flowchart of the pre-sleep routine.

Before the node gets into a low power mode, it goes through a *FreeRTOS* specific routine that occurs right before entering the low power mode, whose flowchart can be seen in figure 4.13. This routine allows doing any necessary coding before the MCU is put in a low power state, and thus it was used to change the peripheral mapping and set the pins as analog inputs for less power consumption (except for the LoRa module *USART* and its enable), and to configure the RTC registers in order to have the correct wake up time. The node then enters the low power mode.

Similarly, *FreeRTOS* has a specific routine that occurs when the node comes out of the low power state, whose flowchart can be see in figure 4.14. Since the RTC is what gives the wake up through its interruption, the node will enter the post sleep subroutine and deactivate the RTC so it won't generate any more interruptions until it enters the next low power cycle. This routine will also restore the peripheral configurations, as it will be making measurements on the sensors.

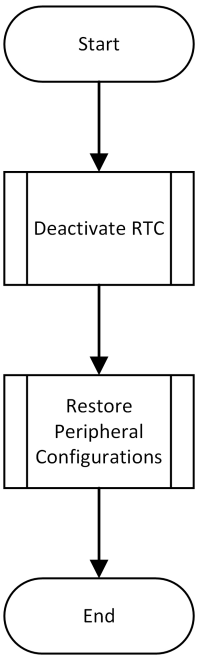


Figure 4.14: Flowchart of the post-sleep routine.

4.3 Conclusion

This chapter described all the development process in order to achieve a final product. It detailed how both nodes were developed, along with their similarities and differences.

Some hardware choices had to be made that differed from the Specification chapter due to restrictions, such as choosing the box, or the board shapes.

Chapter 5

Tests and Results

This chapter contains the tests made to the system. As it was intended to study the LoRa technology and how a LoRaWAN network is built, the tests made reflect that study. It was also necessary to check whether the nodes successfully connected with the upper levels of the network and transmitted information.

5.1 Building an Example LoRaWAN Network

To understand the LoRa modulation and the LoRaWAN protocol, Microchip's *LoRa Technology Evaluation Kit - 800* [76] was used, which contains two example nodes (LoRa Mote), and a gateway. By using this kit, it was intended to observe how a LoRaWAN network works, as the kit provided a *User Guide* which described how to set up the nodes, the gateway, and how to create and run a LoRa server.

The LoRa Mote is a demonstration board which contains an 8-bit PIC18LF45K50 MCU, an RN2483 LoRa module, an LCD display, a light sensor, a temperature sensor, and an AAA battery pack [77]. This node also contains two antenna connectors that allow the use an antenna of a 868 MHz frequency, or one of 433 MHz frequency. The PIC MCU can be programmed through the available In-Circuit Serial Programmer (ICSP). To program this Mote, it was used the PICkit 3 [78].

To program the LoRa Mote, it was used the *MPLAB X* IDE and the C language. The developed code was based on the source code provided by Microchip, which contains an example application for this node. This application had basic commands for the RN2483 module, as well as data collection from the light and temperature sensors integrated in the LoRa Mote.

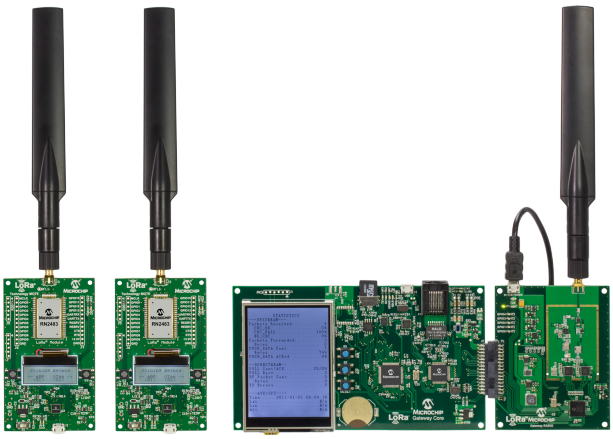


Figure 5.1: Microchip's *LoRa Technology Evaluation Kit - 800*, which contains two LoRa Motes, and a gateway.

When first testing the kit, LoRa Suite guide [79] was followed. The *LoRa Development Utility*, allows the usage of the *LoRa Technology Evaluation Kit - 800*, and provides a Graphical User Interface (GUI) that allows setting up the nodes, the gateway, and the server. It allows to set all the parameters on the nodes and gateway, from the LoRaWAN keys, node Device Address, or the gateway IP.

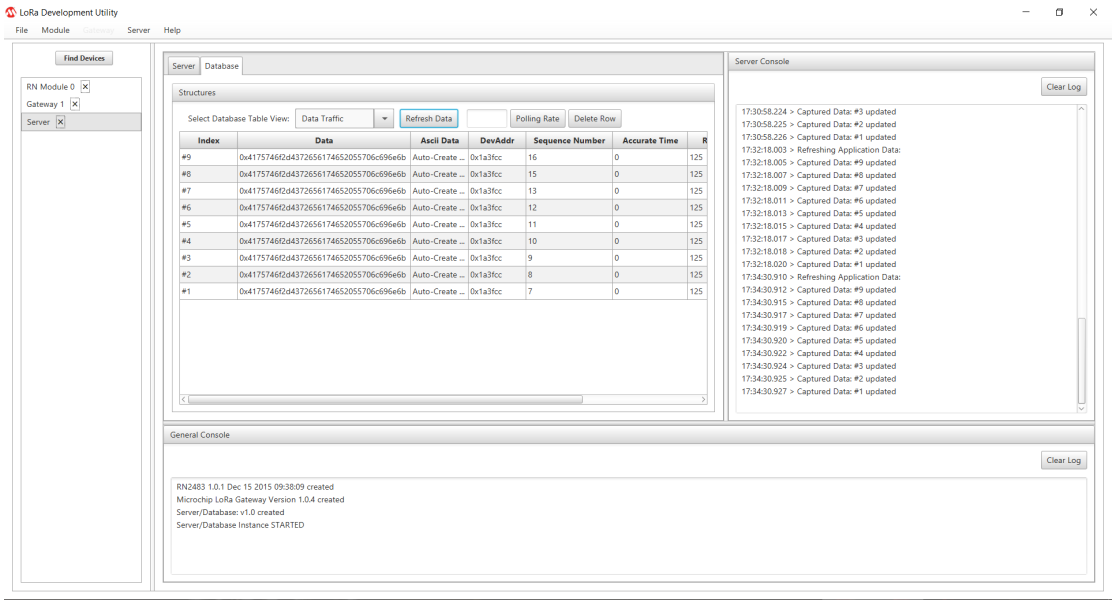


Figure 5.2: *LoRa Development Utility* containing all the connected nodes, gateways, and the LoRa server.

The LoRa server was ran in Docker, which allows virtualization. The server was provided in a Docker Image by Microchip. Although it was possible to properly set up in the *LoRa Development*

Utility the nodes, the gateway, and running the server (as seen in figure 5.2), this software had a lot of issues and did not allow to fully understand how the LoRa technology works, as the messages sent from the nodes could not be seen on the LoRa server. For one, the server was not able to receive the messages from the nodes, even though they could be seen on the gateway traffic. It was only possible to see that the gateway was connected to the server every once in a while, and its status was not consistent.

It was therefore decided to move on to another approach, and use a custom gateways that was developed in parallel with this *Master's thesis* for the purpose of the LPWAN. This was crucial, as the gateway that was used along this project allowed to store and further study the data received by the nodes, whereas Microchip's gateway had a closed firmware and did not allow any sort of changes.

5.2 Transmission Tests

In order to fully understand how the LoRa transmissions worked, a set of tests was done, which consisted of sending messages through the LoRaWAN protocol. These tests took into consideration the different parameters: Spreading Factor, and Data Rate. They are invariably connected, as seen previously on the study made about LoRa. It was important to know how they impacted the transmission time and consumption of the RN2483 module.

The tests done for this used an equivalent resistance of 2.5 in series with the RN2483 module and the oscilloscope probes on the resistance to measure the behaviour of the LoRa module when transmitting through LoRaWAN with different Data Rates. The results have numbering on each section, which makes the following correspondence accordingly:

1. Module is turned off.
2. Moment when the module is turned on, and a serial reply to a small verification command is sent, as well as the serial reply to the ABP.
3. Two serial replies to the commands setting the SF and TxPower parameters.
4. LoRaWAN transmission.

5. Two serial replies, indicating that the LoRaWAN transmission has finished (typically, the node replies with a command acknowledge message, which indicates whether it got a valid command or not, and a message indicating whether the message transmission was successful).

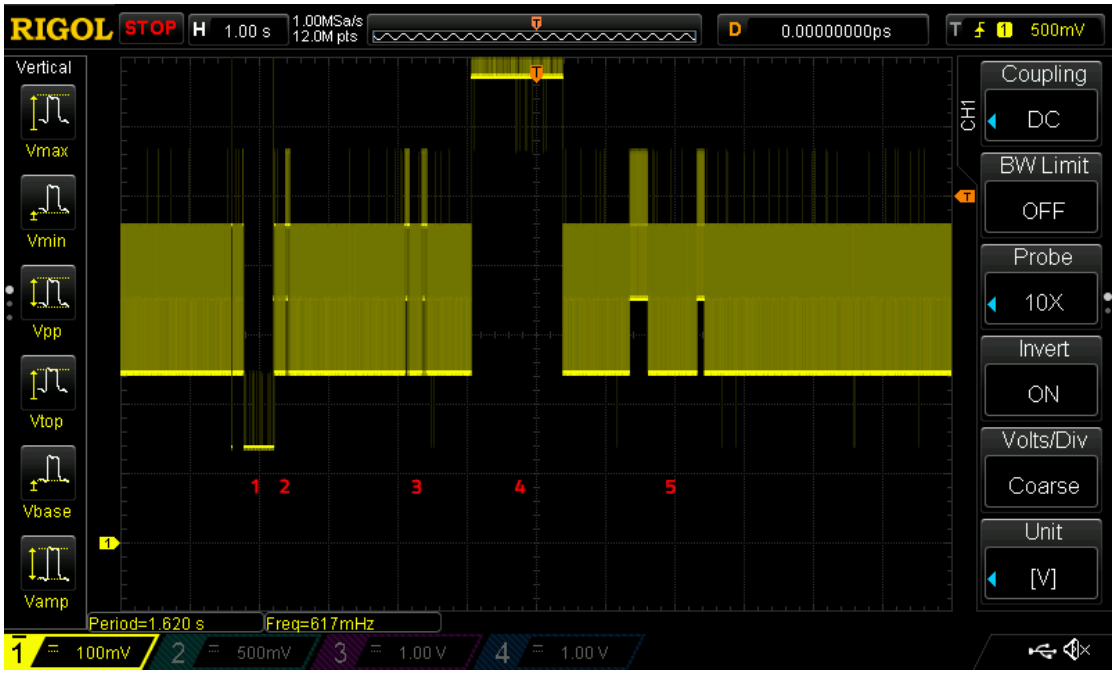


Figure 5.3: First transmission test. Data Rate was set to 0, thus the SF value was 7.

Three Data Rates/Spreading Factor values were chosen, which were the highest (DR5/SF12) and lowest value (DR0/SF7), and a middle range value (DR2/SF10).

The first test (figure 5.3) was done with the Data Rate set to 0, which corresponds to the lowest SF value, SF 7. This means that the message was transmitted at the bit rate of 250 bps. This is the lowest speed at which the LoRaWAN module can transmit. As expected, the message transmission took a longer time than in the other tests, taking approximately more than 1 second.

The second test (figure 5.4) had the Data Rate set to 2, which corresponds to a Spreading Factor value SF10. Hence, the message was transmitted at a bit rate of 980 bps. Compared to the first test, it is possible to see that the transmission time is relatively shorter.

The third test (figure 5.5) had a Data Rate value of 5, which is the highest possible value for this parameter. This set the bit rate at 5470 bps, and as seen in by the result, the transmission is the fastest when compared to the other tests.

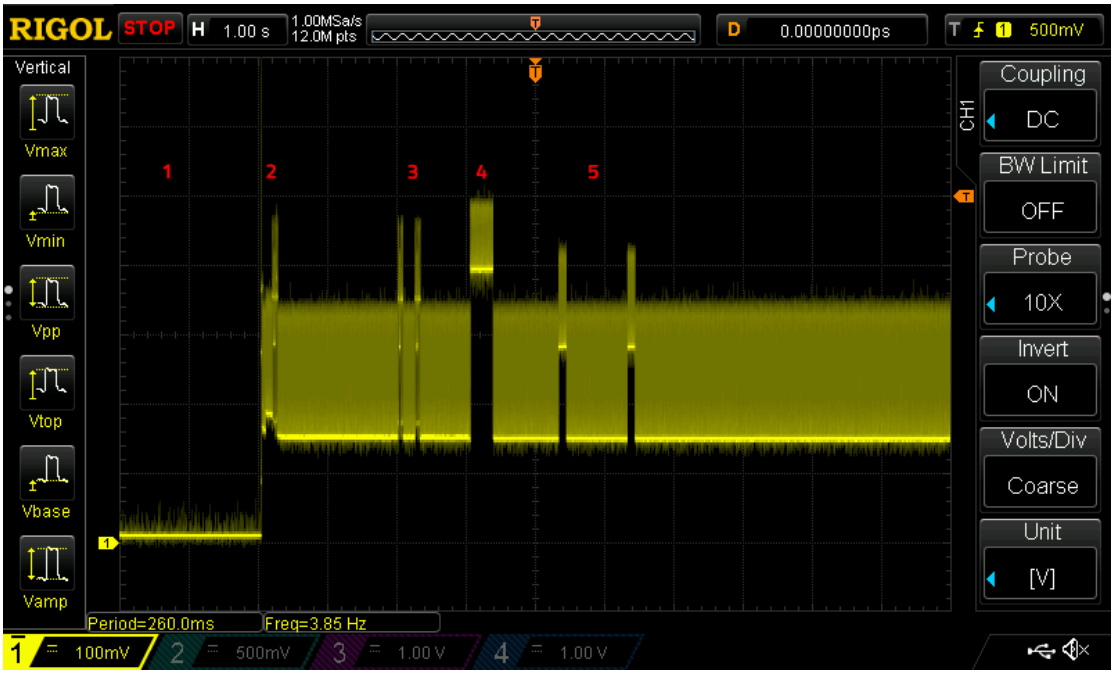


Figure 5.4: Second transmission test. Data Rate was set to 2, thus the SF value was 10.

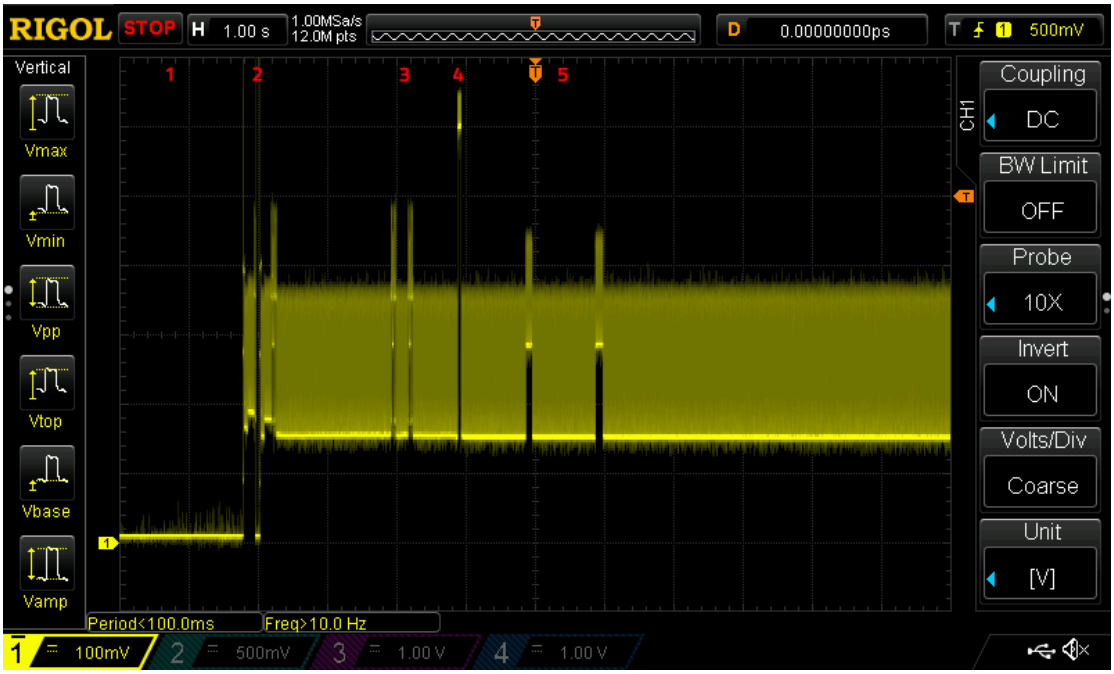


Figure 5.5: Third transmission test. Data Rate was set to 5, thus the SF value was 12.

Despite being able to get all the data transmission, there was a lot of noise on the signals. This was due to the power source that emitted a lot of noise itself, and even adding a capacitor to create a filter did not seem to make much of an impact. However, it was still possible to capture and analyse the results.

5.3 Network connection

After understanding how the LoRa transmissions work, it was proceeded to test the connectivity with the LoRaWAN server and how the messages would be viewed from the Application Server's point of view. The Application Server used the was The Things Network (TTN), which contains also the Network Server.

After successfully joining the device with the LoRaWAN network, and integrating it with the TTN network server, it was sent one simple message to see how the application handled it, and what kind of information it could retrieve from it. The payload content was kept simple.


```
{
"app_id":"1323232321",
"dev_id":"1223",
"hardware_serial":"0000000000000109",
"port":5,
"counter":55,
"payload_raw":"AAAA",
"metadata":
  {
  "time":"2018-11-16T17:16:46.214146755Z",
  "frequency":868.3,
  "modulation":"LORA",
  "data_rate":"SF12BW125",
  "coding_rate":"4/5",
  "gateways":
  [
    {
    "gtw_id":"eui-aa555a0402230101",
    "timestamp":488802892,
    "time":"",
    "channel":1,
    "rssi":-109,
    "snr":-8,
    "rf_chain":1,
    "latitude":41.551163,
    "longitude":-8.375217,
    "location_source":"registry"
    }
  ]
  },
"downlink_url":"https://integrations.thethingsnetwork.org/ttn-eu/api/v2/down
/1323232321/1243?key=ttn-account-v2.KqQRkF63WDF2_VmHRrA8QiI8h8KCEr6b0PVnnUyaP9g"
}
```

Listing 5.1: Test message sent from a node as seen in the Application Server

Listing 5.1 shows the information received from the Network/Application server. Since the TTN does both the Network and Application server, it is able to decode the payload, as the application keys are generated by it.

Apart from the node's keys, and payload information, it is also possible to see the LoRa parameters configuration. Furthermore, all data regarding the gateway is also shown, along with the gateway's location coordinates.

Therefore, it was possible to establish a connection between the nodes and the Application Server, by means of a gateway.

5.4 Range Tests

Two sets of tests were done to try and obtain the maximum range the LoRa module can provide to the nodes and how accurate its transmissions are. It was also intended to verify how the different parameters (Spreading Factor and Transmission Power) would affect the communications, as well as how the Network Server and the Application Server would handle data.

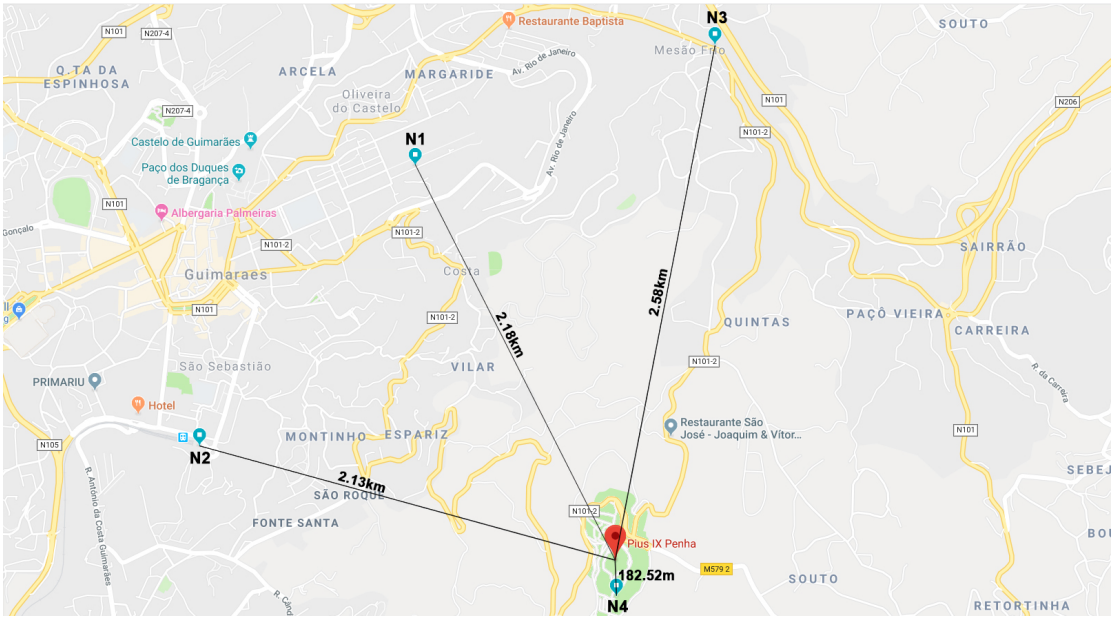


Figure 5.6: Map with Gateway and Nodes locations for the first set of tests.

The first set of tests was done in the urban city environment, in Guimarães. The gateway was placed in a stationary position at the highest possible point, in Penha. It was used one node, which was moved around the city. For each set of tests on a given location, it was considered as if it was a different node for an easier mapping and explanation of the different results. The Network Server used was TTN, and all the application data was sent to a custom Application Server, which received and handled all the received messages.

Figure 5.6 shows the map with the gateway and the nodes' locations for the first set of tests. Table 5.1 shows how far each node was for the gateway, and whether it had a line of sight or not. To better understand and know the LoRa capabilities, the nodes were placed in different locations with distinct properties, such as line of sight to the gateway, or the amount of buildings around it. The coordinates and elevation for each node and the gateway can be found in table 5.2.

Table 5.1: Nodes' distance to the Gateway and line of sight for the first set of tests.

Node	Distance to gateway	Line of sight
Node 1	2.18 km	Partial
Node 2	2.13 km	Yes
Node 3	2.58 km	No
Node 4	182.52 m	No

Table 5.2: Nodes and Gateway geolocation for the first set of tests.

Device	Latitude	Longitude	Elevation
Node 1	41,447191	-8,280558	215 m
Node 2	41,434758	-8,293337	208 m
Node 3	41,452558	-8,262819	331 m
Node 4	41,428094	-8,268661	590 m
Gateway	41,429672	-8,26854	603 m

In all the tests that were made, the Spreading Factor (SF) and Transmission Power (TxPower) varied from all the available combinations of values for each, to obtain different results and further understand how these two parameters can influence the success rate of message transmission.

As previously mentioned in section 2.4, LoRaWAN supports six different Spreading Factors (SF7-SF12). The RN2483 allows five different TxPower values (Txp1 - Txp5). For each SF value, starting from the highest (SF12) to the lowest (SF7), three messages combining the SF values and all the available values of TxPower were used, also from the highest (Txp1) to the lowest (Txp5). Two tests were done per node, with a total of 90 messages sent for each test.

The results of the tests made can be found in table 5.3, where the percentage of packages received by each node can be see, for each different SF value.

Table 5.3: Received Rate Percentage for each node for the first set of tests.

Node	SF7	SF8	SF9	SF10	SF11	SF12
N1	0%	3,3%	16,6%	30%	46,7%	43,3%
N2	93,3%	50%	60%	50%	50%	66,7%
N3	0%	0%	0%	0%	0%	0%
N4	3,3%	13,3%	26,7%	30%	30%	43,3%

The gateway was placed on the highest possible point. For the first location, the node had a partial line of sight to the gateway location, as it was surrounded by a couple of trees in a park. It was observed that for the highest DR values (or lowest SF values), the results were quite poor, as most of the messages did not arrive correctly. This can probably be explained due to the line of sight not being 100% clear, as well as the distance itself playing a role for the highest Data Rate values.

The second node had a clear line of sight to the place where the gateway was, and it was on a straight line ahead of it, on a lower spot. These settings may have proven crucial for the high percentage of packages received by the gateway when the transmission occurred on the highest Data Rate. The results for this node were the best among all the tests, most likely due to the favourable conditions mentioned previously.

The third node was located on the furthest position of all the tests. The line of sight towards the gateway was also non existent, as there was buildings around it, and it was on the back side of the position where the gateway was located. These poor placement conditions explain why none of the messages were delivered, on either test.

The fourth node was the closest to the gateway. However, it was still below it, and it did not

have a line of sight to the gateway, as there were rock formations in between. Hence, the results were not very positive, and were lower than expected for such a close location.

The second set of tests was also done in the urban city environment, in Braga. The gateway was placed in a stationary position at a high altitude point, in Bom Jesus. Two nodes were used, one which was stationary, and the other was moved around the city. Similarly to the first test, all the different locations were marked as if they were different nodes.

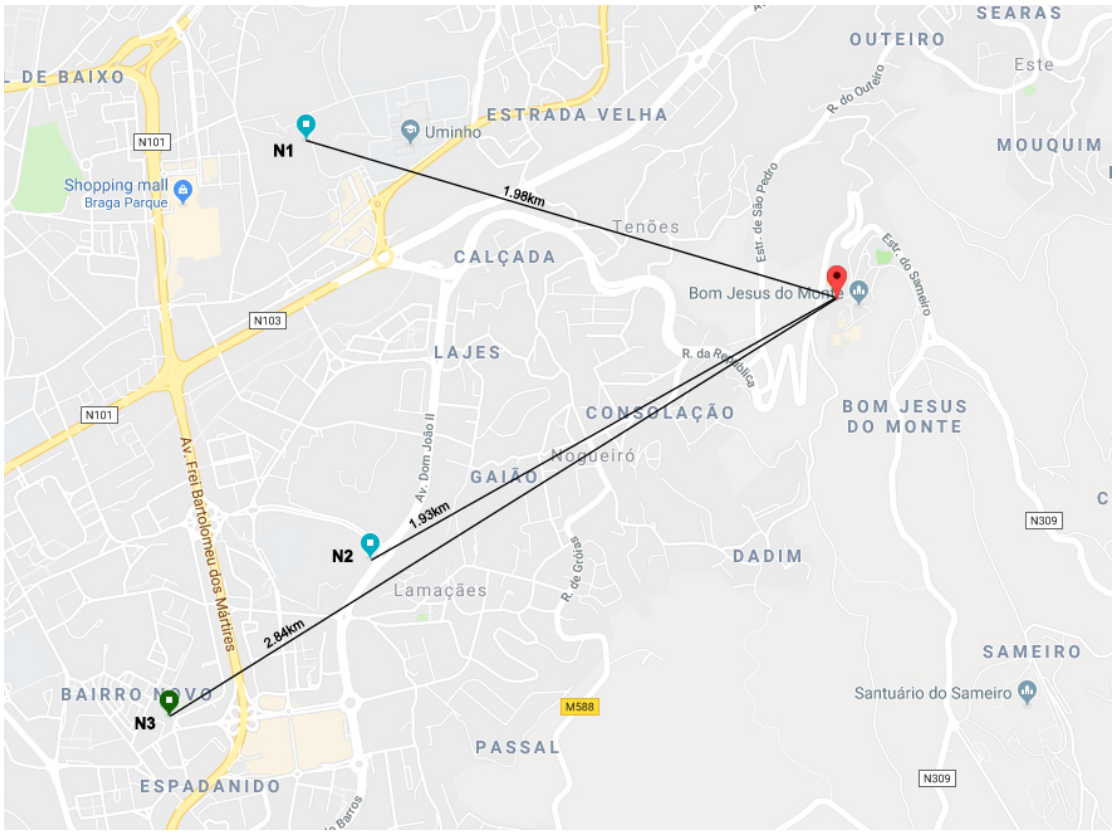


Figure 5.7: Map with Gateway and Nodes locations for the second set of tests.

Figure 5.7 shows the map with the gateway and the nodes' locations for the second set of tests. Table 5.4 shows how far each node was for the gateway, and whether it had a line of sight or not. The coordinates and elevation for each node and the gateway can be found in table 5.5.

Table 5.4: Nodes' distance to the Gateway and line of sight for the second set of tests.

Node	Distance to gateway	Line of sight
Node 1	1.93 km	Yes
Node 2	1.98 km	Yes

Continues on the following page

Table 5.4 – Continuation of the previous page

Node	Distance to gateway	Line of sight
Node 3	2.84 km	Yes

The testing method was similar to the first test, with the Spreading Factor (SF) and Transmission Power (TxPower). The SF values varied from SF7 to SF12, whereas the TxPower values used were only three (Txp1, Txp3 and Txp5). For each SF value, starting from the highest (SF12) to the lowest (SF7), one hundred messages were sent combining the SF values and the three TxPower, from highest (Txp1) to the lowest (Txp5) values. One test was made per location, for the node that was moved from one place to the other, with a total of 1800 messages sent for each test. The stationary node (N3) was continuously transmitting.

Table 5.5: Nodes and Gateway geolocation for the second set of tests.

Device	Latitude	Longitude	Elevation
Node 1	41,560078	-8,400911	189 m
Node 2	41,546564	-8,398164	205 m
Node 3	41,541495	-8,406812	203 m
Gateway	41,555111	-8,378013	381 m

The results of the tests made can be found in table 5.6, where the percentages of packages received by each node can be seen for each different SF value.

Table 5.6: Received Rate Percentage for each node for the second set of tests.

Node	SF7	SF8	SF9	SF10	SF11	SF12
N1	85%	90%	100%	100%	100%	100%
N2	80%	85%	90%	100%	100%	100%
N3	0%	0%	0%	0%	0%	0%

The gateway was placed at Bom Jesus. There was a possibility of placing it in Sameiro, which is a higher point. However, it was decided not to do so because Sameiro contains several antennas near its location, which could play a high role on causing interference with the message reception. Since the location of the gateway on the first set of tests already had this factor, as Penha contains

antennas, it was important to see how the results would be affected without having this factor. The results obtained were significantly better, even considering that the nodes were located at a longer distance from the gateway than in the first set of tests. The lowest percentage of messages received was for the highest DR values (lower SF values).

The first node was located on a parking lot, which had a direct line of sight to the gateway location, on a lower elevation. From the moving nodes, this was the closest. Most of the messages sent by this node were successful, especially for the lower DR values (higher SF parameter value). For the higher DR values, there were some messages that were not received, which was expected for such a distance in an urban area.

The second node was also located in a parking lot with direct line of sight. It was on a lower elevation than the gateway, although a bit higher than the first node. This was the furthest node from the mobile nodes, and its results were also successful for the lower DR values. For the lower DR values (SF7, SF8, and SF9), there was some packet loss that was slightly higher than in the previous node, which may have been due to the distance effect.

The third node was the furthest away for both tests. This node was placed on the fifth floor of a balcony, with a direct line of sight to the gateway location, and was constantly transmitting the messages as it was cycling through the different parameter combinations. Comparing its results to the results obtained in the previous test, in which no message was received from the furthest node, it could be concluded that distance was the main factor that prevent the message reception, as non of the messages were delivered.

The Received Signal Strength Indication and Signal-to-Noise Ratio graphics from both the first test and the second test can be seen in Appendix C and Appendix D respectively.

Chapter 6

Conclusions and Future Work

After the whole implementation of the nodes, conclusions can be made from the developed work and the results obtained. This chapter aims to make an overview of the results obtained and what can be concluded from them, as well as suggest further improvements that can be made in future implementations.

6.1 Conclusion

The conclusions are based on all the tests made to both variants of the nodes. The LoRa module was tested extensively, in order to have an in-depth comprehension of how this protocol actually works, its limitations, and maximum range that can be achieved. It was also possible to see how the different parameters could affect the transmissions, which has an impact on the energy consumption of the nodes. The authentication with the Network and Application server was successful, and it was possible to send data to the upper levels. However, due to restrictions from the TTN regarding the downlink messages, it was not possible to test the confirmed messages.

LoRa proved to be effective for long range transmissions, as the results obtained in an urban city environment showed that it is possible to send message within the distance of kilometres. However, more tests should be done in remote areas, such as vast forests, as the city environment contains a lot more interference than remote areas. The position of the gateway and the nodes is also an important factor for the success rate, as objects in-between can interfere with the quality of the signal. The gateway should also be set on a higher location than the nodes, as this proved to be a good approach.

It was also tested the possibility of having a node transmit to another. However, this solution did not seem viable. Even though it was possible to send and receive messages from node to node,

the LoRa module some limitations to receiving messages. Firstly, it can only receive one message at a time, having either a continuous receive mode, or a limited time for message reception. This means that the nodes would have to be well coordinated in order to send messages between them. As they receive messages from any node that is on the same frequency, there is a high probability of a node receive an unintended message, while missing a message it was supposed to receive. Furthermore, to send this type of message, it is only using the LoRa modulation and not the LoRaWAN protocol, which contains no security or encryption.

As it was intended to do some first studies into the LoRa technology and create a node architecture, there was only two nodes developed for this concept, and thus it was not possible to make extensive tests regarding how the network would handle multiple communications at the same time. In order to do this, a larger number of nodes would be required, and syncing them all to send several messages.

One of the biggest difficulties in reducing the production price for these nodes is the cost of gas sensors. As the most accurate sensors can be very expensive, it is necessary to either chose accuracy or a lower costs. However, it will not be as precise, and therefore a compromise should be made between accuracy, and cost. However, if using other types of sensors, this should not be an issue.

6.2 Future Work

For an upcoming version of the nodes, the board size can be further reduced to the enclosure that was mentioned in the Specification chapter. This box is actually the one mostly used for the research group's project, and thus would fit the standard sizing that is usually used for projects that involve sensor nodes. An external RTC should also be added, as the internal one of the MCU is not very precise.

There were some difficulties with implementing the lowest power modes along with FreeRTOS. Although it was possible to implement all the low power modes, with FreeRTOS this was only done with the *Sleep* mode, which was not the ideal mode. Finding a solution to this would be ideal for the nodes' low power consumption.

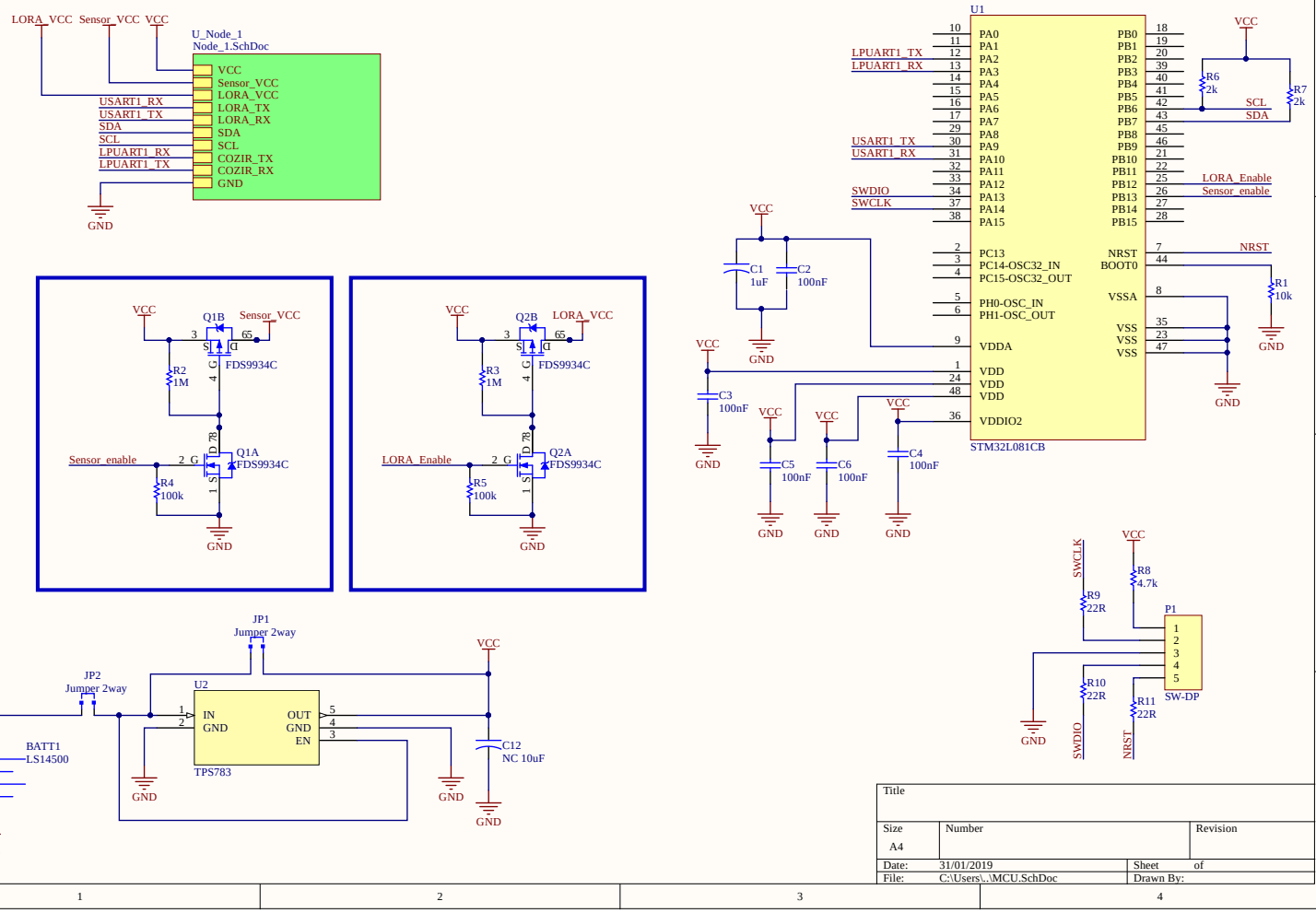
It would be crucial to test both nodes and LoRa's effectiveness in a real application scenario, which would mean leaving both the nodes and the gateway outdoors. The ideal test would be

in an area where firefighters could make use of the system in their training exercises, and this would not only help to prove the system's functionality and intuitiveness for emergency services, but also to see how the system would react in a real world scenario. It would also provide the needed data for the upper layers to be able to make predictions on fire occurrences.

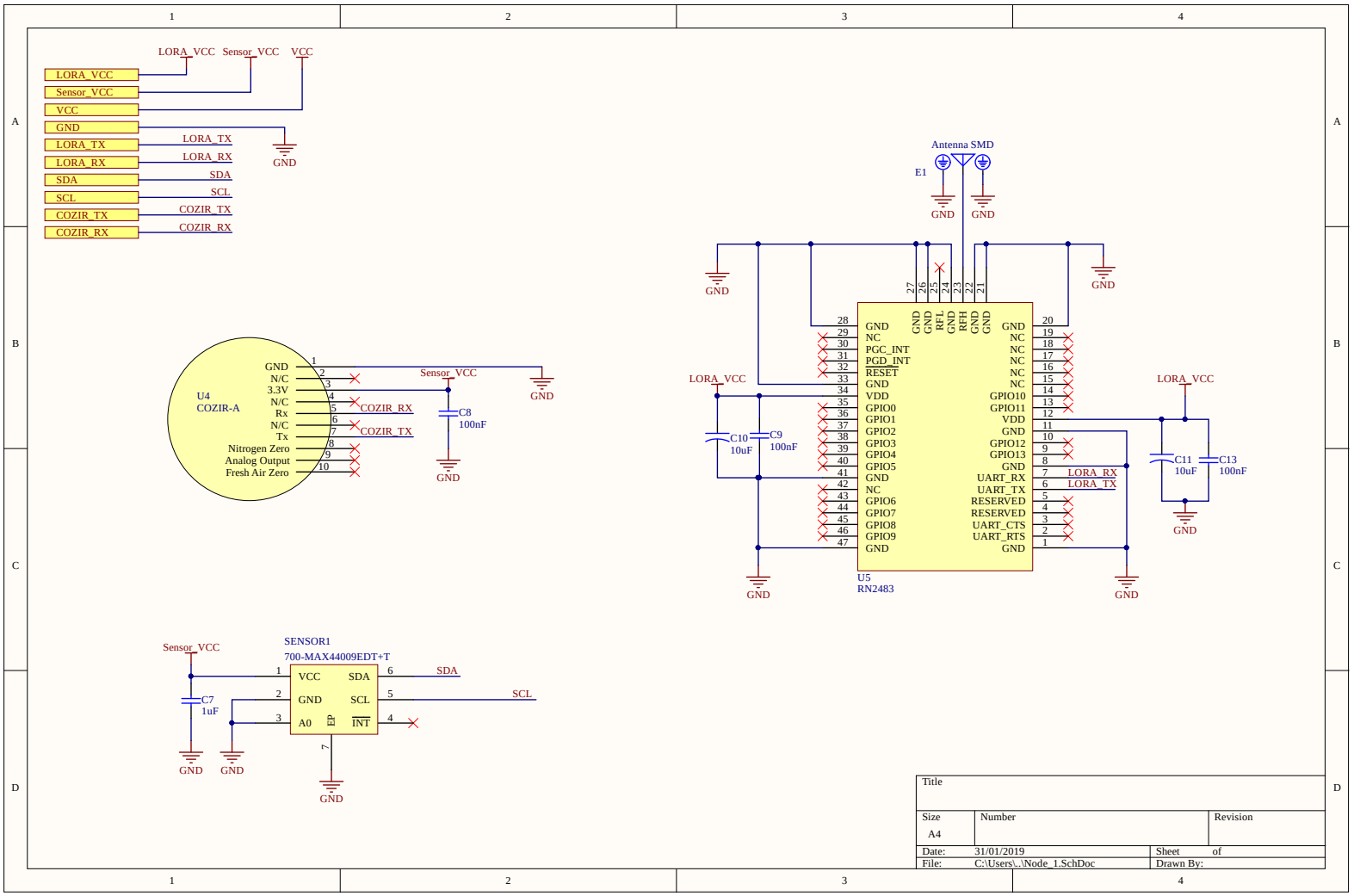
Lastly, it should also be tested the confirmed data messages, as it was implemented but not tested. Ideally, it would be better to use a custom Application Server, as having information dependent on a third party might not be a good approach for sensible data. This could be done by doing some encryption on the payload, and since the chosen microcontroller supports AES encryption, it is possible to make a full integration of the encryption keys on both the nodes and the Application Server.

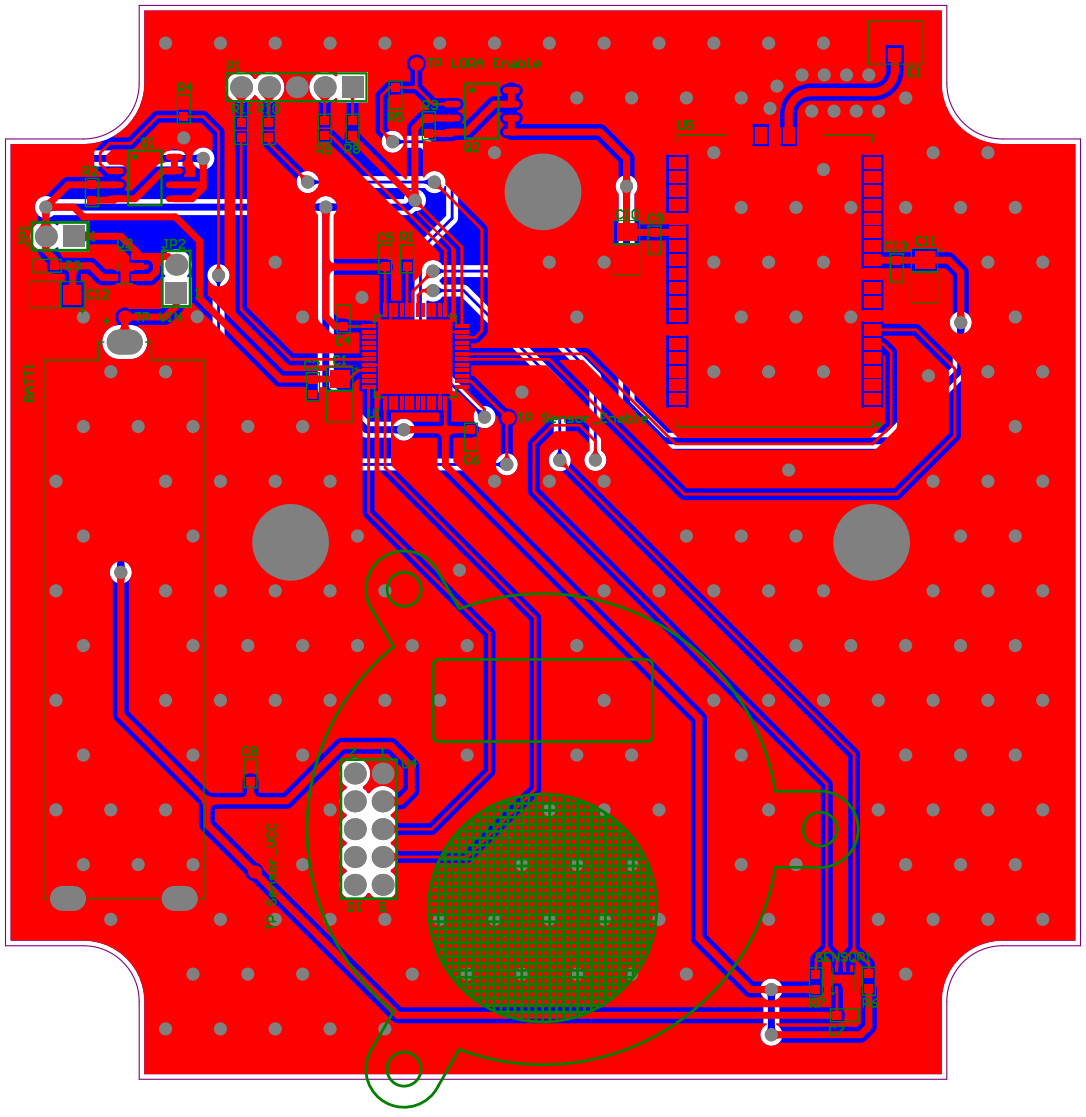
Appendix A

Variant 1 Schematics and Layout



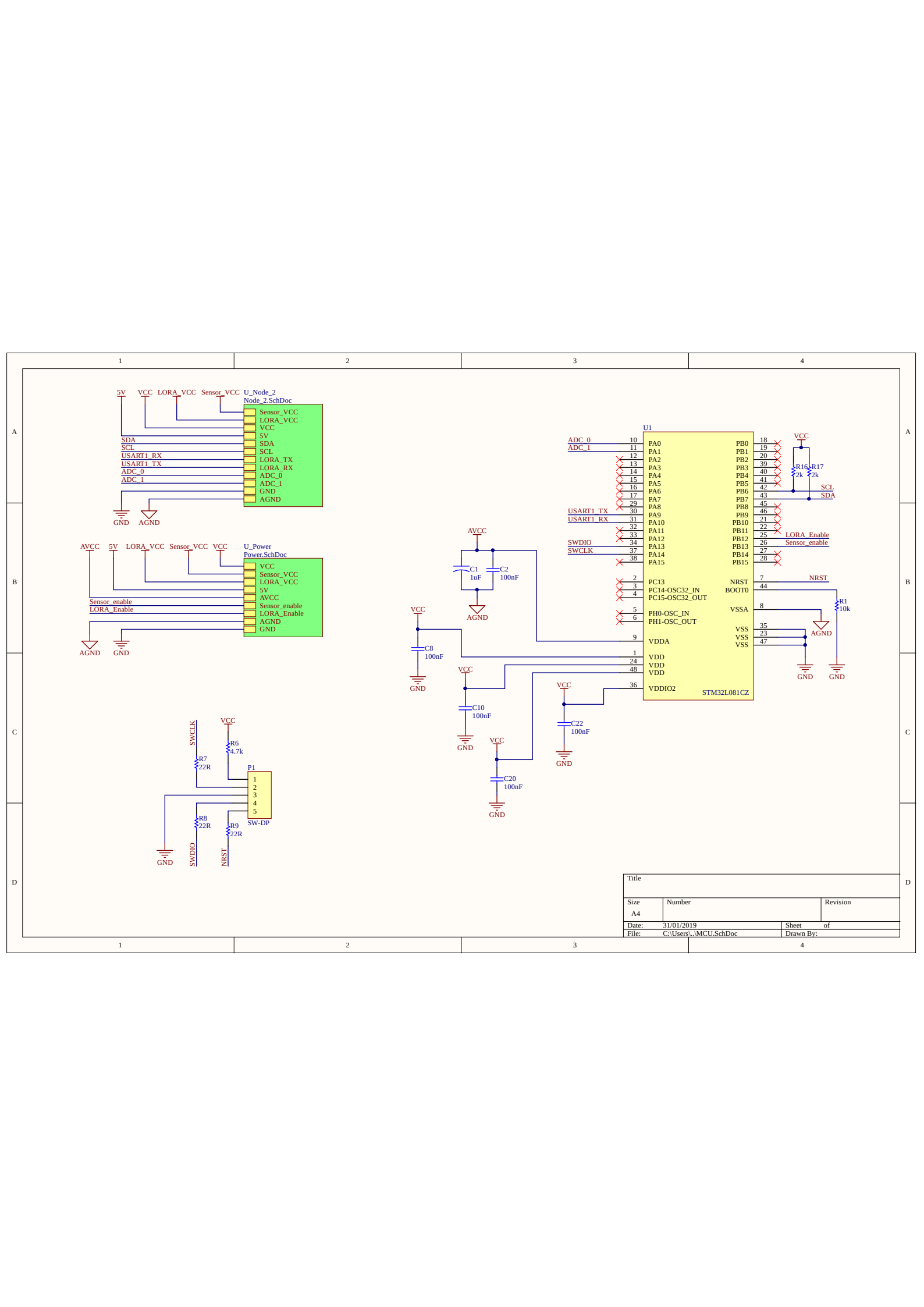
Title		
Size: A4	Number	Revision
Date: 31/01/2019		Sheet of
File: C:\Users\..\MCU.SchDoc		Drawn By:

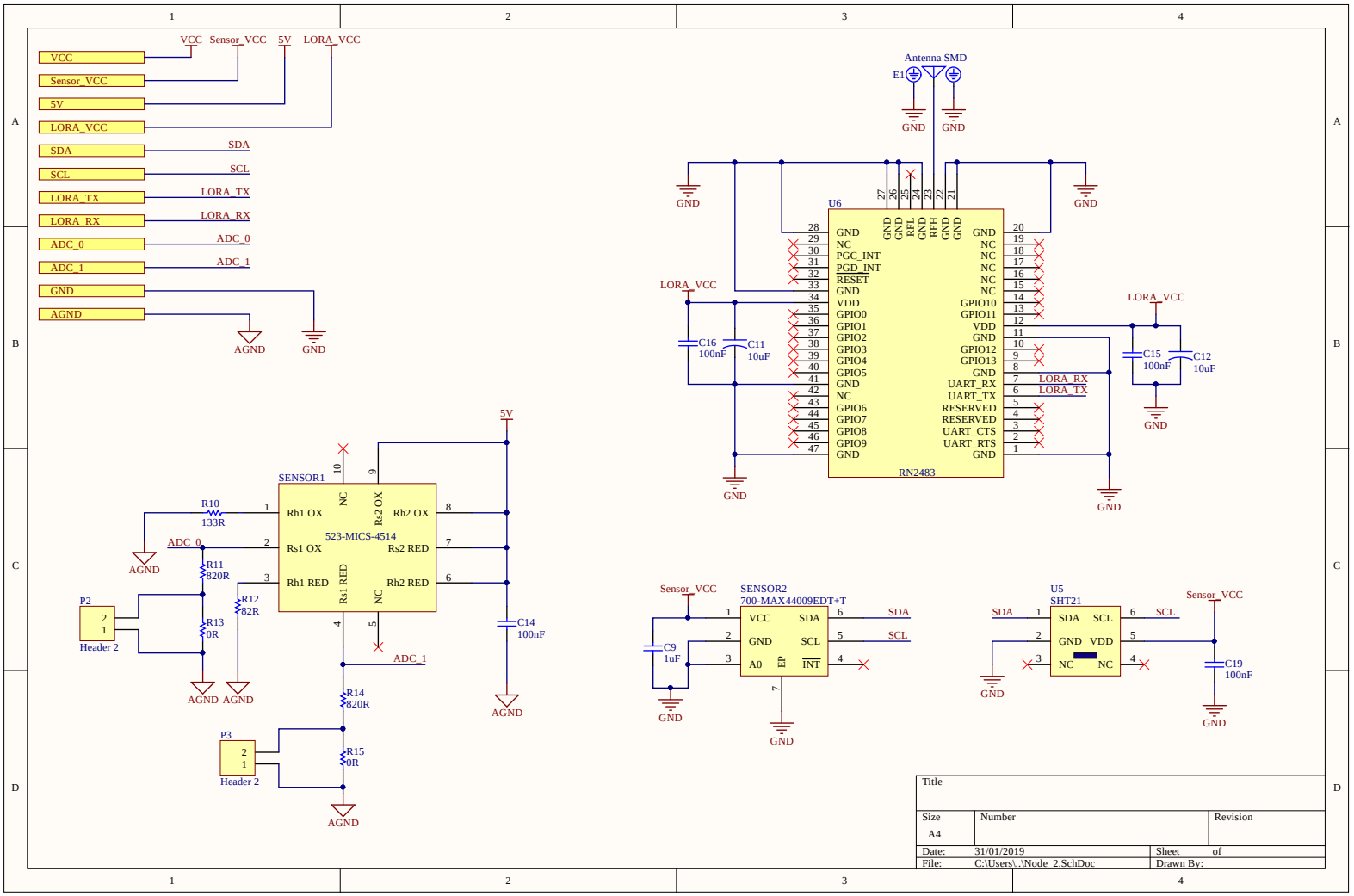

LORA_VCC
Sensor_VCC
VCC
GND
LORA_TX
LORA_RX
SDA
SCL
COZIR_TX
COZIR_RX
U4
COZIR-A
Nitrogen Zero
Analog Output
Fresh Air Zero
C8
100nF
Antenna SMD
E1
U5
RN2483
C9
100nF
C10
10uF
C11
10uF
C13
100nF
SENSOR1
700-MAX44009EDT+T
C7
1uF
VCC
SDA
GND
SCL
A0
INT
Title
Size
A4
Number
Revision
Date:
31/01/2019
Sheet
of
File:
C:\Users\..\Node_1.SchDoc
Drawn By:

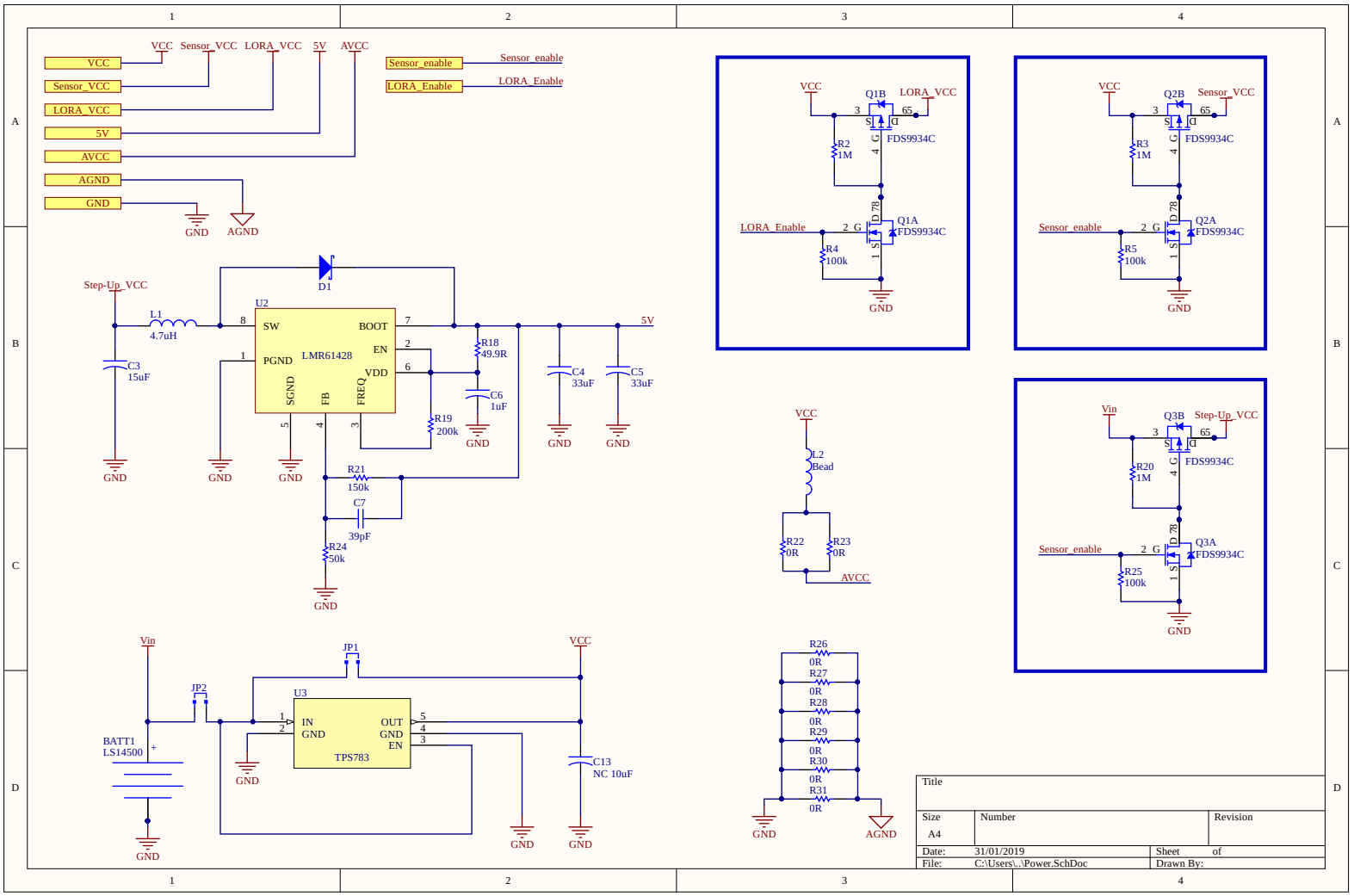

TP LORA Enable
TP Sensor Enable
BATT1
Power_VCC
U5
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29
R30
R31
R32
R33
R34
R35
R36
R37
R38
R39
R40
R41
R42
R43
R44
R45
R46
R47
R48
R49
R50
R51
R52
R53
R54
R55
R56
R57
R58
R59
R60
R61
R62
R63
R64
R65
R66
R67
R68
R69
R70
R71
R72
R73
R74
R75
R76
R77
R78
R79
R80
R81
R82
R83
R84
R85
R86
R87
R88
R89
R90
R91
R92
R93
R94
R95
R96
R97
R98
R99
R100
R101
R102
R103
R104
R105
R106
R107
R108
R109
R110
R111
R112
R113
R114
R115
R116
R117
R118
R119
R120
R121
R122
R123
R124
R125
R126
R127
R128
R129
R130
R131
R132
R133
R134
R135
R136
R137
R138
R139
R140
R141
R142
R143
R144
R145
R146
R147
R148
R149
R150
R151
R152
R153
R154
R155
R156
R157
R158
R159
R160
R161
R162
R163
R164
R165
R166
R167
R168
R169
R170
R171
R172
R173
R174
R175
R176
R177
R178
R179
R180
R181
R182
R183
R184
R185
R186
R187
R188
R189
R190
R191
R192
R193
R194
R195
R196
R197
R198
R199
R200
R201
R202
R203
R204
R205
R206
R207
R208
R209
R210
R211
R212
R213
R214
R215
R216
R217
R218
R219
R220
R221
R222
R223
R224
R225
R226
R227
R228
R229
R230
R231
R232
R233
R234
R235
R236
R237
R238
R239
R240
R241
R242
R243
R244
R245
R246
R247
R248
R249
R250
R251
R252
R253
R254
R255
R256
R257
R258
R259
R260
R261
R262
R263
R264
R265
R266
R267
R268
R269
R270
R271
R272
R273
R274
R275
R276
R277
R278
R279
R280
R281
R282
R283
R284
R285
R286
R287
R288
R289
R290
R291
R292
R293
R294
R295
R296
R297
R298
R299
R300
R301
R302
R303
R304
R305
R306
R307
R308
R309
R310
R311
R312
R313
R314
R315
R316
R317
R318
R319
R320
R321
R322
R323
R324
R325
R326
R327
R328
R329
R330
R331
R332
R333
R334
R335
R336
R337
R338
R339
R340
R341
R342
R343
R344
R345
R346
R347
R348
R349
R350
R351
R352
R353
R354
R355
R356
R357
R358
R359
R360
R361
R362
R363
R364
R365
R366
R367
R368
R369
R370
R371
R372
R373
R374
R375
R376
R377
R378
R379
R380
R381
R382
R383
R384
R385
R386
R387
R388
R389
R390
R391
R392
R393
R394
R395
R396
R397
R398
R399
R400
R401
R402
R403
R404
R405
R406
R407
R408
R409
R410
R411
R412
R413
R414
R415
R416
R417
R418
R419
R420
R421
R422
R423
R424
R425
R426
R427
R428
R429
R430
R431
R432
R433
R434
R435
R436
R437
R438
R439
R440
R441
R442
R443
R444
R445
R446
R447
R448
R449
R450
R451
R452
R453
R454
R455
R456
R457
R458
R459
R460
R461
R462
R463
R464
R465
R466
R467
R468
R469
R470
R471
R472
R473
R474
R475
R476
R477
R478
R479
R480
R481
R482
R483
R484
R485
R486
R487
R488
R489
R490
R491
R492
R493
R494
R495
R496
R497
R498
R499
R500
R501
R502
R503
R504
R505
R506
R507
R508
R509
R510
R511
R512
R513
R514
R515
R516
R517
R518
R519
R520
R521
R522
R523
R524
R525
R526
R527
R528
R529
R530
R531
R532
R533
R534
R535
R536
R537
R538
R539
R540
R541
R542
R543
R544
R545
R546
R547
R548
R549
R550
R551
R552
R553
R554
R555
R556
R557
R558
R559
R560
R561
R562
R563
R564
R565
R566
R567
R568
R569
R570
R571
R572
R573
R574
R575
R576
R577
R578
R579
R580
R581
R582
R583
R584
R585
R586
R587
R588
R589
R590
R591
R592
R593
R594
R595
R596
R597
R598
R599
R600
R601
R602
R603
R604
R605
R606
R607
R608
R609
R610
R611
R612
R613
R614
R615
R616
R617
R618
R619
R620
R621
R622
R623
R624
R625
R626
R627
R628
R629
R630
R631
R632
R633
R634
R635
R636
R637
R638
R639
R640
R641
R642
R643
R644
R645
R646
R647
R648
R649
R650
R651
R652
R653
R654
R655
R656
R657
R658
R659
R660
R661
R662
R663
R664
R665
R666
R667
R668
R669
R670
R671
R672
R673
R674
R675
R676
R677
R678
R679
R680
R681
R682
R683
R684
R685
R686
R687
R688
R689
R690
R691
R692
R693
R694
R695
R696
R697
R698
R699
R700
R701
R702
R703
R704
R705
R706
R707
R708
R709
R710
R711
R712
R713
R714
R715
R716
R717
R718
R719
R720
R721
R722
R723
R724
R725
R726
R727
R728
R729
R730
R731
R732
R733
R734
R735
R736
R737
R738
R739
R740
R741
R742
R743
R744
R745
R746
R747
R748
R749
R750
R751
R752
R753
R754
R755
R756
R757
R758
R759
R760
R761
R762
R763
R764
R765
R766
R767
R768
R769
R770
R771
R772
R773
R774
R775
R776
R777
R778
R779
R780
R781
R782
R783
R784
R785
R786
R787
R788
R789
R790
R791
R792
R793
R794
R795
R796
R797
R798
R799
R800
R801
R802
R803
R804
R805
R806
R807
R808
R809
R810
R811
R812
R813
R814
R815
R816
R817
R818
R819
R820
R821
R822
R823
R824
R825
R826
R827
R828
R829
R830
R831
R832
R833
R834
R835
R836
R837
R838
R839
R840
R841
R842
R843
R844
R845
R846
R847
R848
R849
R850
R851
R852
R853
R854
R855
R856
R857
R858
R859
R860
R861
R862
R863
R864
R865
R866
R867
R868
R869
R870
R871
R872
R873
R874
R875
R876
R877
R878
R879
R880
R881
R882
R883
R884
R885
R886
R887
R888
R889
R890
R891
R892
R893
R894
R895
R896
R897
R898
R899
R900
R901
R902
R903
R904
R905
R906
R907
R908
R909
R910
R911
R912
R913
R914
R915
R916
R917
R918
R919
R920
R921
R922
R923
R924
R925
R926
R927
R928
R929
R930
R931
R932
R933
R934
R935
R936
R937
R938
R939
R940
R941
R942
R943
R944
R945
R946
R947
R948
R949
R950
R951
R952
R953
R954
R955
R956
R957
R958
R959
R960
R961
R962
R963
R964
R965
R966
R967
R968
R969
R970
R971
R972
R973
R974
R975
R976
R977
R978
R979
R980
R981
R982
R983
R984
R985
R986
R987
R988
R989
R990
R991
R992
R993
R994
R995
R996
R997
R998
R999
R1000

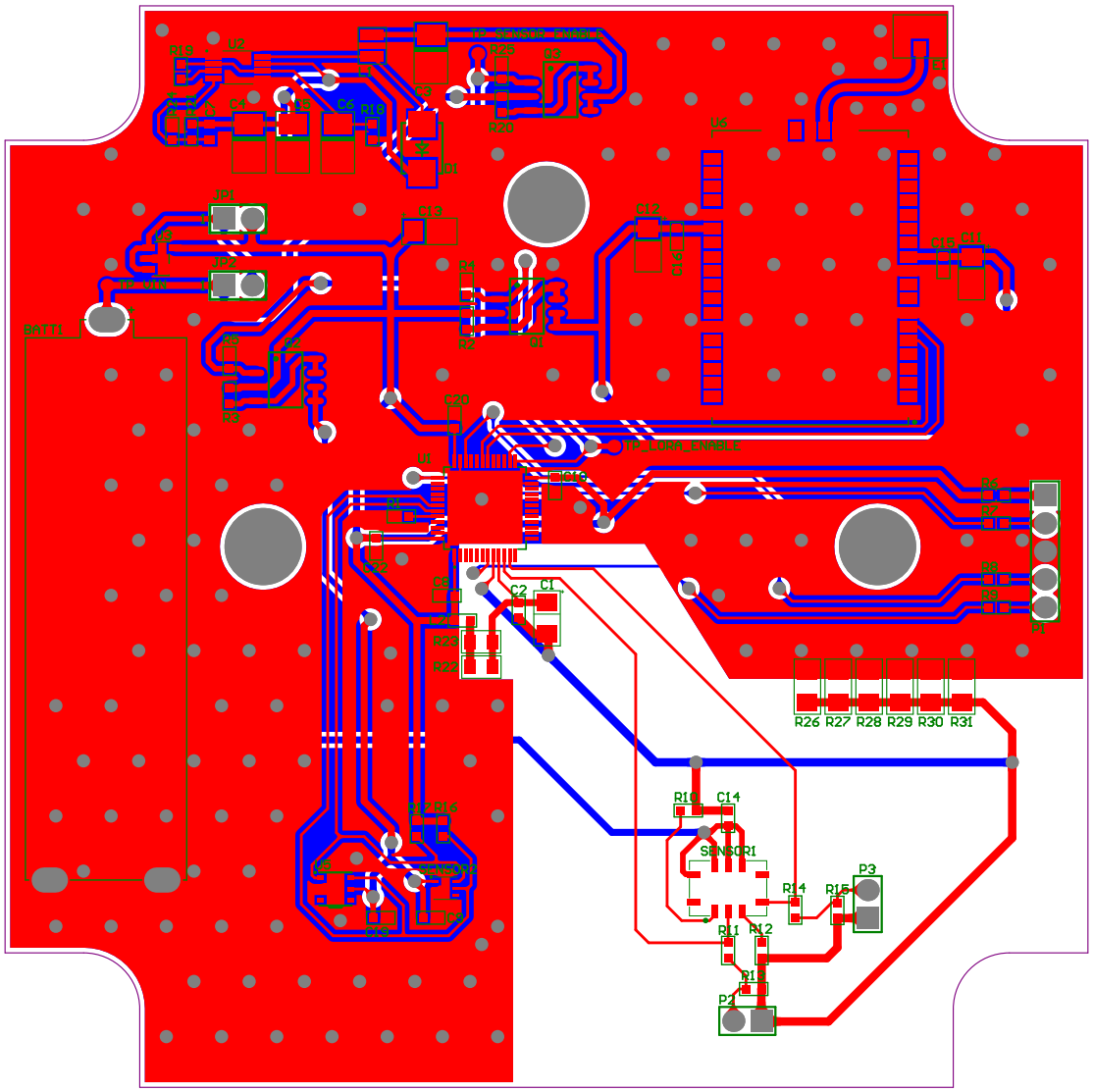
Appendix B

Variant 2 Schematics and Layout


5V
VCC
LORA_VCC
Sensor_VCC
U_Node_2
Node_2.SchDoc
Sensor_VCC
LORA_VCC
VCC
5V
SDA
SCL
LORA_TX
LORA_RX
ADC_0
ADC_1
GND
AGND
SDA
SCL
USART1_RX
USART1_TX
ADC_0
ADC_1
GND
AGND
AVCC
5V
LORA_VCC
Sensor_VCC
VCC
U_Power
Power.SchDoc
VCC
Sensor_VCC
LORA_VCC
5V
AVCC
Sensor_enable
LORA_Enable
AGND
GND
Sensor_enable
LORA_Enable
AGND
GND
U1
ADC_0
ADC_1
USART1_TX
USART1_RX
SWDIO
SWCLK
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PC13
PC14-OSC32_IN
PC15-OSC32_OUT
PH0-OSC_IN
PH1-OSC_OUT
VDDA
VDD
VDD
VDD
VDDIO2
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
BOOT0
VSSA
VSS
VSS
VSS
STM32L081CZ
VCC
R16
2k
R17
2k
SCL
SDA
LORA_Enable
Sensor_enable
NRST
R1
10k
AGND
GND
GND
AVCC
C1
1uF
C2
100nF
AGND
VCC
C8
100nF
GND
VCC
C10
100nF
GND
VCC
C22
100nF
GND
VCC
C20
100nF
GND
SWCLK
VCC
R6
4.7k
R7
22R
P1
1
2
3
4
5
SW-DP
R8
22R
R9
22R
GND
SWDIO
NRST
Title
Size
A4
Number
Revision
Date:
31/01/2019
Sheet
of
File:
C:\Users\..\MCU.SchDoc
Drawn By:


VCC
Sensor_VCC
5V
LORA_VCC
SDA
SCL
LORA_TX
LORA_RX
ADC_0
ADC_1
GND
AGND
Antenna SMD
E1
U6
RN2483
SENSOR1
523-MICS-4514
SENSOR2
700-MAX44009EDT+T
U5
SHT21
P2
Header 2
P3
Header 2
R10
133R
R11
820R
R12
82R
R13
0R
R14
820R
R15
0R
C9
1uF
C11
10uF
C12
10uF
C14
100nF
C15
100nF
C16
100nF
C19
100nF
Title
Size
A4
Number
Revision
Date:
31/01/2019
Sheet
of
File:
C:\Users\..\Node_2.SchDoc
Drawn By:

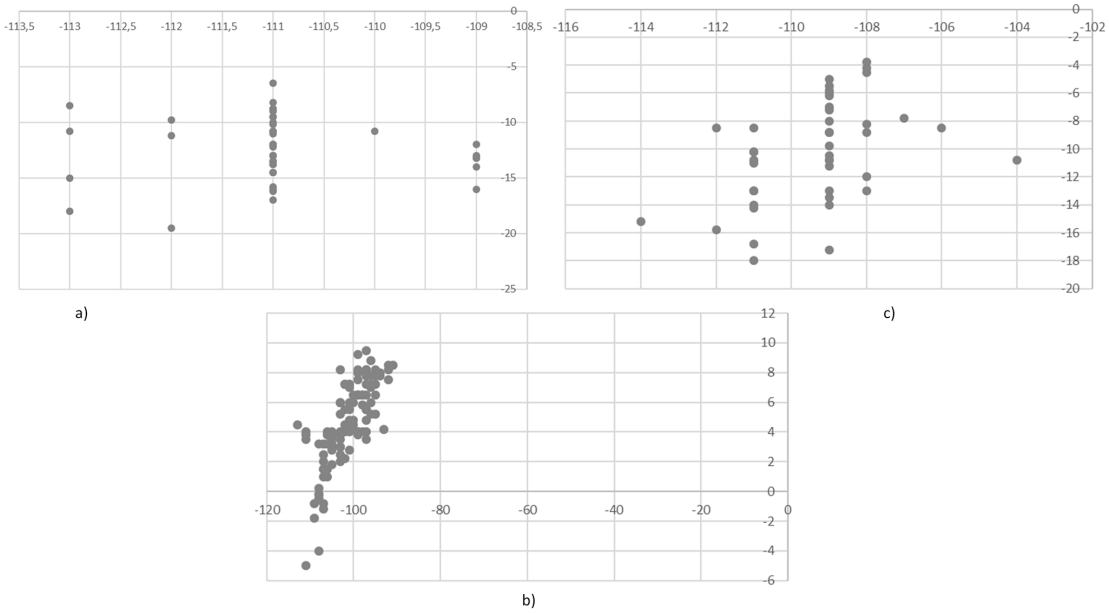

VCC
Sensor_VCC
LORA_VCC
5V
AVCC
AGND
GND
Sensor_enable
LORA_Enable
Step-Up_VCC
D1
L1
4.7uH
U2
SW
BOOT
PGND
LMR61428
EN
VDD
SGND
FB
FREQ
C3
15uF
R18
49.9R
C4
33uF
C5
33uF
C6
1uF
R19
200k
R21
150k
C7
39pF
R24
50k
Q1B
Q1A
FDS9934C
R2
1M
R4
100k
Q2B
Q2A
R3
1M
R5
100k
Q3B
Q3A
R20
1M
R25
100k
L2
Bead
R22
0R
R23
0R
R26
0R
R27
0R
R28
0R
R29
0R
R30
0R
R31
0R
Vin
JP1
JP2
U3
IN
OUT
TPS783
BATT1
LS14500
C13
NC 10uF
Title
Size
A4
Number
Revision
Date:
31/01/2019
Sheet
of
File:
C:\Users\..\Power.SchDoc
Drawn By:


U2
R19
C4
JP1
JP2
BATT1
C13
C12
C11
C15
R4
R2
Q1
C20
U1
C5
C2
C1
R23
R22
R6
R7
R8
R9
D1
R26 R27 R28 R29 R30 R31
R10
C14
SENSOR1
P3
R14
R15
R11
R12
R13
P2
R16

Appendix C

Received Signal Strength Indication and Signal-to-Noise Ratio First Test

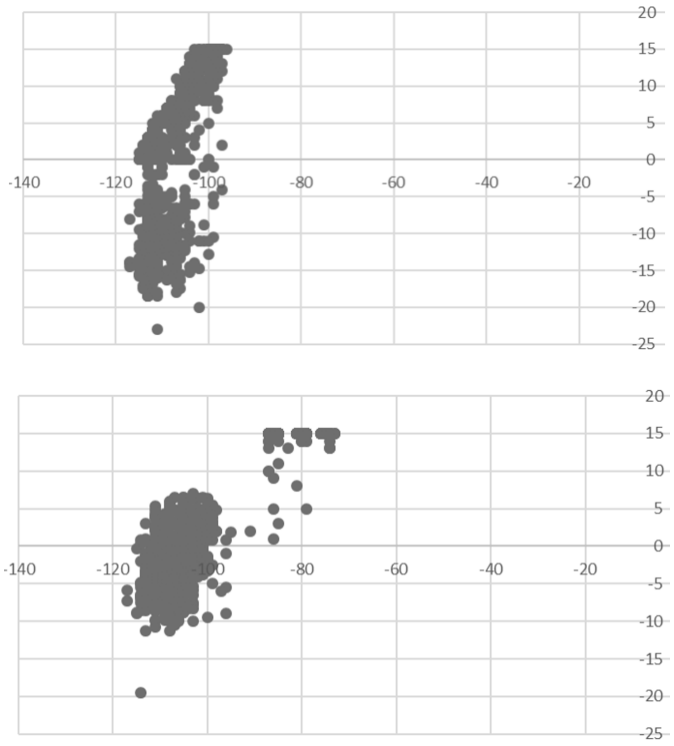
The following image displays the graphics of the Signal-to-Noise Ratio (SNR) (vertical axis) and the Received Signal Strength Indication (RSSI) (horizontal axis) obtained from the first set of tests. Graphic a) is relative to Node 1, graphic b) is relative to Node 2, and graphic c) is relative to Node 4.



Appendix D

Received Signal Strength Indication and Signal-to-Noise Ratio Second Test

The following image displays the graphics of the Signal-to-Noise Ratio (SNR) (vertical axis) and the Received Signal Strength Indication (RSSI) (horizontal axis) obtained from the second set of tests. The top graphic is relative to Node 2, and the bottom graphic is relative to Node 1.



References

[1] N. Jazdi, "Cyber physical systems in the context of Industry 4.0," in *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*. IEEE, may 2014, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/6857843/

[2] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An Overview on Wireless Sensor Networks Technology and Evolution," *Sensors*, vol. 9, pp. 6869–6896, 2009. [Online]. Available: www.mdpi.com/journal/sensors

[3] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7815384/

[4] A. A. A. Alkhatib, "A Review on Forest Fire Detection Techniques," *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, p. 597368, mar 2014. [Online]. Available: http://journals.sagepub.com/doi/10.1155/2014/597368

[5] S. Ornes, "Core Concept: The Internet of Things and the explosion of interconnectivity." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, no. 40, pp. 11 059–11 060, oct 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/27702874http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5056067

[6] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, vol. 3, no. 3, pp. 164–173, 2015. [Online]. Available: http://dx.doi.org/10.4236/jcc.2015.35021

[7] S. P. Urbanski, W. M. Hao, and S. Baker, "Chemical Composition of Wildland Fire Emissions," *Developments in Environmental Science*, vol. 8. [Online]. Available: https://www.fs.fed.us/rm/pubs_other/rmrs_2009_urbanski_s001.pdf

[8] Edward A. Johnson and Kiyoko Miyanishi, *Forest Fires: Behavior and Ecological Effects*. Elsevier, 2001. [Online]. Available: https://books.google.co.in/books/about/Forest_Fires.html?id=MXa8npbbahQC&pgis=1&hl=pt-PT

[9] "Tudo o que se sabe sobre "o pior dia de incêndios do ano" – Observador." [Online]. Available: http://observador.pt/2017/10/16/tudo-o-que-se-sabe-sobre-o-pior-dia-de-incendios-do-ano/ [Accessed: 2017-11-28]

[10] "Relatório da Comissão Técnica Independente," 2017. [Online]. Available: https://www.parlamento.pt/Paginas/2017/outubro/CT-Independente-analise-incendios.aspx [Accessed: 2017-10-24]

[11] International Electrotechnical Commission, "Internet of Things: Wireless Sensor Networks." [Online]. Available: http://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf

[12] S. Kumar and D. Shepherd, "SensIT: Sensor information technology for the warfighter," *Proc. 4th Int. Conf. on Information Fusion*, no. January 2001, pp. 1–7, 2001. [Online]. Available: http://isif.org/fusion/proceedings/fusion01CD/fusion/searchengine/pdf/TuC11.pdf

[13] E. Cayirci, R. Govindan, T. Znati, and M. Srivastava, "Wireless sensor networks," *Computer Networks*, vol. 43, no. 4, pp. 417–419, nov 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128603003517?via%3Dihub

[14] S. Kumar Gupta and P. Sinha, "Overview of Wireless Sensor Network: A Survey," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 1, 2014. [Online]. Available: https://ijarcce.com/wp-content/uploads/2012/03/IJARCCE7D__a_sanjeev_overview.pdf

[15] R. Baheti and H. Gill, *Cyber-Physical Systems: From Theory to Practice*, 2011, no. 1.

[16] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems," in *Proceedings of the 47th Design Automation Conference on - DAC '10*. New York, New York, USA: ACM Press, 2010, p. 731. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1837274.1837461

[17] R. Poovendran, “Cyber–Physical Systems: Close Encounters Between Two Parallel Worlds [Point of View],” *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1363–1366, aug 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5512708/

[18] A. Hellinger and H. Seeger, *Cyber-Physical Systems - Driving force for innovation in mobility, health, energy and production*, acatech – National Academy of Science and Engineering, Ed., 2011.

[19] “LPWAN - The Benefits of LPWAN Technology vs. Other IoT Connectivity Options | IoT For All.” [Online]. Available: https://www.iotforall.com/lpwan-benefits-vs-iot-connectivity-options/ [Accessed: 2017-09-11]

[20] “I-ON Communications Blog: LPWA (Low Power Wide Area), the core of IoT.” [Online]. Available: https://ioncomm.blogspot.com/2016/10/lpwa-low-power-wide-area-core-of-iot.html [Accessed: 2017-12-13]

[21] Ericsson AB, “5G radio access – capabilities and technologies.” [Online]. Available: https://www.ericsson.com/assets/local/publications/white-papers/wp-5g.pdf

[22] “SigFox Vs. LoRa: A Comparison Between Technologies & Business Models.” [Online]. Available: https://www.link-labs.com/blog/sigfox-vs-lora [Accessed: 2018-08-08]

[23] “How Sigfox plans to spread its low-power IoT network across the U.S.” [Online]. Available: https://www.networkworld.com/article/3029253/internet-of-things/how-sigfox-plans-to-spread-its-low-power-iot-network-across-the-u-s.html [Accessed: 2017-09-14]

[24] “Sigfox Technology Overview | Sigfox.” [Online]. Available: https://www.sigfox.com/en/sigfox-iot-technology-overview [Accessed: 2017-09-11]

[25] Semtech, “Smart Cities Transformed.” [Online]. Available: https://www.semtech.com/lora/resources/lora-white-papers

[26] “About LoRaWAN™ | LoRa Alliance™.” [Online]. Available: https://lora-alliance.org/about-lorawan [Accessed: 2018-08-10]

[27] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14–21, mar 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959517300061?via%3Dihub

[28] "Network Architecture | The Things Network." [Online]. Available: https://www.thethingsnetwork.org/docs/network/architecture.html [Accessed: 2018-09-10]

[29] "What is LoRaWAN ? - ResIOT LoRaWAN Network Server and IoT Platform." [Online]. Available: https://www.resiot.io/en/what-is-lorawan/ [Accessed: 2018-08-23]

[30] "lora-alliance | Technology." [Online]. Available: https://www.lora-alliance.org/technology [Accessed: 2017-09-07]

[31] LoRa Alliance, "LoRaWAN™ 1.1 Specification," *LoRaWAN™ 1.1 Specification*, no. 2017, p. 97331, 2017. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf

[32] "Adaptive Data Rate | The Things Network." [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/adr.html [Accessed: 2018-09-04]

[33] "A technical overview of LoRa® and LoRaWAN ™," 2015. [Online]. Available: https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf

[34] "RPMA | u-blox." [Online]. Available: https://www.u-blox.com/en/rpma [Accessed: 2017-09-18]

[35] "RPMA — Overview of Ingenu's LPWAN Technology – IoT For All – Medium." [Online]. Available: https://medium.com/iotforall/rpma-overview-of-ingenus-lpwan-technology-3d72c47f0461 [Accessed: 2017-09-18]

[36] "Narrowband – Internet of Things (NB-IoT) | Internet of Things." [Online]. Available: https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/ [Accessed: 2017-11-24]

[37] Y. D. Beyene, R. Jantti, O. Tirkkonen, K. Ruttik, S. Iraji, A. Larmo, T. Tirronen, Torsner, and Johan, "NB-IoT Technology Overview and Experience from Cloud-RAN Implementation,"

IEEE Wireless Communications, vol. 24, no. 3, pp. 26–32, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7955908/

[38] Semtech Corporation, "SX1276/77/78/79 Datasheet," no. August, 2016. [Online]. Available: https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V5.pdf

[39] LoRa Alliance, "LoRa Specification." [Online]. Available: https://lora-alliance.org/sites/default/files/2018-05/2015_-_lorawan_specification_1r0_611_1.pdf

[40] "Spreading Factor (SF), Time on Air and (Adaptive) Data Rate | Zakelijk KPN Forum." [Online]. Available: https://zakelijkforum.kpn.com/lora-forum-16/spreading-factor-sf-time-on-air-and-adaptive-data-rate-10908 [Accessed: 2019-01-02]

[41] "Nitrogen dioxide (NO2) - Air quality fact sheet." [Online]. Available: http://www.environment.gov.au/protection/publications/factsheet-nitrogen-dioxide-no2 [Accessed: 2018-04-16]

[42] NASA, "Nitrogen Dioxide." [Online]. Available: https://eospso.gsfc.nasa.gov/sites/default/files/publications/NO2poster_508.pdf

[43] "Sources of Nitrogen Oxides - Chemistry LibreTexts." [Online]. Available: https://chem.libretexts.org/Core/Environmental_Chemistry/Acid_Rain/Sources_of_Nitrogen_Oxides [Accessed: 2018-04-16]

[44] "Fire & Carbon Monoxide," 2018. [Online]. Available: https://earthobservatory.nasa.gov/global-maps/MOD14A1_M_FIRE/MOP_CO_M [Accessed: 2018-10-24]

[45] "Carbon dioxide." [Online]. Available: https://www.sciencedaily.com/terms/carbon_dioxide.htm [Accessed: 2018-10-26]

[46] B. Kleven, "A Summary of Gas Detection," no. 520, 2016. [Online]. Available: https://www.goacd.com/downloads/documentation/gas-det-summary.pdf

[47] G. Korotcenkov, *Handbook of Gas Sensor Materials*, ser. Integrated Analytical Systems. New York, NY: Springer New York, 2014, vol. 1. [Online]. Available: http://link.springer.com/10.1007/978-1-4614-7388-6

[48] "Gas Detection Handbook." [Online]. Available: http://www.gilsoneng.com/reference/gasdetectionhandbook.pdf

[49] "Measuring air pollution with low-cost sensors." [Online]. Available: http://ec.europa.eu/environment/air/pdf/Brochurelower-costsensors.pdf

[50] P. J. D. Peterson, A. Aujla, K. H. Grant, A. G. Brundle, M. R. Thompson, J. V. Hey, and R. J. Leigh, "Practical Use of Metal Oxide Semiconductor Gas Sensors for Measuring Nitrogen Dioxide and Ozone in Urban Environments," 2017. [Online]. Available: https://pdfs.semanticscholar.org/55f5/f8e0a0b425f55b9f53eaff881da0f6babbf6.pdf

[51] G. F. Fine, L. M. Cavanagh, A. Afonja, and R. Binions, "Metal Oxide Semi-Conductor Gas Sensors in Environmental Monitoring," *Sensors*, vol. 10, no. 6, pp. 5469–5502, jun 2010. [Online]. Available: http://www.mdpi.com/1424-8220/10/6/5469

[52] M. Aleixandre and M. Gerboles, "Review of Small Commercial Sensors for Indicative Monitoring of Ambient Gas." [Online]. Available: http://www.aidic.it/cet/12/30/029.pdf

[53] X. Liu, S. Cheng, H. Liu, S. Hu, D. Zhang, and H. Ning, "A Survey on Gas Sensing Technology," *Sensors*, vol. 12, no. 7, pp. 9635–9665, jul 2012. [Online]. Available: http://www.mdpi.com/1424-8220/12/7/9635

[54] "Types of Temperature Sensors | DigiKey." [Online]. Available: https://www.digikey.pt/en/blog/types-of-temperature-sensors [Accessed: 2018-10-29]

[55] "4 Most Common Types of Temperature Sensor | Ametherm." [Online]. Available: https://www.ametherm.com/blog/thermistors/temperature-sensor-types [Accessed: 2018-10-29]

[56] "Absolute vs. Relative Humidity - What's the Difference?" [Online]. Available: http://info.zehnderamerica.com/blog/absolute-vs.-relative-humidity-whats-the-difference [Accessed: 2018-11-04]

[57] "Choosing a Humidity Sensor: A Review of Three Technologies | Sensors Magazine." [Online]. Available: https://www.sensorsmag.com/components/choosing-a-humidity-sensor-a-review-three-technologies [Accessed: 2018-10-31]

[58] National Electrical Manufacturers Association, "ANSI/IEC 60529-2004 Degrees of Protection Provided by Enclosures (IP Code)," pp. 1–8, 2004. [Online]. Available: https://www.nema.org/Standards/ComplimentaryDocuments/ANSI-IEC-60529.pdf

[59] Sealite, "IP - Ingress Protection Rating," 2016. [Online]. Available: https://www.sealite.com/wp-content/uploads/2018/01/2_pdf.pdf

[60] "STM32L0 - ARM Cortex-M0+ ultra-low-power MCUs - STMicroelectronics." [Online]. Available: https://www.st.com/en/microcontrollers/stm32l0-series.html?querycriteria=productId=SS1817 [Accessed: 2018-12-04]

[61] "COZIR Datasheet," p. 3. [Online]. Available: http://www.co2meters.com/Documentation/Datasheets/DS-GC-0010-COZIR-Ambient.pdf

[62] "RN2483 LoRa® Transceiver Module - Microchip Technology | Mouser Portugal." [Online]. Available: https://pt.mouser.com/new/microchip/microchip-rn2483-module/ [Accessed: 2019-01-07]

[63] Microchip, "RN2483 Datasheet," pp. 1–22, 2017. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/50002346C.pdf

[64] "RN2483 - Wireless Modules - Microcontrollers and Processors." [Online]. Available: https://www.microchip.com/wwwproducts/en/RN2483 [Accessed: 2018-10-08]

[65] SAFT, "Primary lithium battery LS 14500," no. September, pp. 7–8, 2009. [Online]. Available: https://tinyurl.com/y94ctdcz

[66] "1554E2GYSL Hammond Manufacturing | Boxes, Enclosures, Racks | DigiKey." [Online]. Available: https://www.digikey.pt/products/en/boxes-enclosures-racks/boxes/594?k=UL+94+V-0&k=&pkeyword=UL+94+V-0&sv=0&pv46=19077&sf=0&FV=ffec17a6%2Cffe00252&quantity=&ColumnSort=0&page=1&pageSize=25 [Accessed: 2018-11-05]

[67] "1554EGY Hammond Manufacturing | Boxes, Enclosures, Racks | DigiKey." [Online]. Available: https://www.digikey.pt/product-detail/en/hammond-manufacturing/1554EGY/HM920-ND/1090732 [Accessed: 2018-11-05]

[68] "Venting | Automotive, Packaging, Protective & Portable Electronics Vents | Gore." [Online]. Available: https://www.gore.com/products/categories/venting [Accessed: 2018-11-05]

[69] "VENT-PS1YBK-N8001 Amphenol LTW | Mouser Portugal." [Online]. Available: https://pt.mouser.com/ProductDetail/Amphenol-LTW/VENT-PS1YBK-N8001?qs=sGAEpiMZZMve4%2FbfQkoj%252bDkooasw%252b%252bjU5MJ0s0M12gc%3D [Accessed: 2018-11-05]

[70] "STM32CubeMX - STM32Cube initialization code generator - STMicroelectronics." [Online]. Available: https://www.st.com/en/development-tools/stm32cubemx.html [Accessed: 2019-01-15]

[71] "NUCLEO-L073RZ - STM32 Nucleo-64 development board with STM32L073RZ MCU, supports Arduino and ST morpho connectivity - STMicroelectronics." [Online]. Available: https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo/nucleo-l073rz.html [Accessed: 2019-01-15]

[72] "ST-LINK/V2 - ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32 - STMicroelectronics." [Online]. Available: https://www.st.com/en/development-tools/st-link-v2.html [Accessed: 2019-01-15]

[73] "STSW-LINK004 - STM32 ST-LINK utility - STMicroelectronics." [Online]. Available: https://www.st.com/en/development-tools/stsw-link004.html [Accessed: 2019-01-15]

[74] ST, "User manual Developing Applications on STM32Cube with RTOS," no. June, pp. 1–26, 2014. [Online]. Available: https://www.st.com/content/ccc/resource/technical/document/user_manual/2d/60/ff/15/8c/c9/43/77/DM00105262.pdf/files/DM00105262.pdf/jcr:content/translations/en.DM00105262.pdf

[75] ——, "STM32L0xx ultra-low power features overview," vol. AN4445, no. February, pp. 1–17, 2014. [Online]. Available: https://www.st.com/content/ccc/resource/technical/document/application_note/27/58/8e/81/79/fb/4f/ac/DM00108286.pdf/files/DM00108286.pdf/jcr:content/translations/en.DM00108286.pdf

[76] "LoRa(R) Technology Evaluation Kit - 800." [Online]. Available: http://www.microchip.com/DevelopmentTools/ProductDetails/dv164140-1#additional-summary [Accessed: 2018-10-08]

[77] Microchip, *LoRa Mote User's Guide*, 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/LoRaMoteUsersGuide.pdf

[78] "PICkit™ 3 In-Circuit Debugger." [Online]. Available: https://www.microchip.com/Developmenttools/ProductDetails/PG164130 [Accessed: 2019-01-03]

[79] *LoRa ® Technology Evaluation Suite User ' s Guide*, 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/40001847A.pdf